Smart Tour Guide

Faculty of Artificial Intelligence

STG Team

Smart Tour Guide

Submitted by:

Mahmoud Hamza Abdella

Abdelraouf Abdelmonem Abdelhakeem

Ola Emad Naseem Zahran

Alaa Taha Muhammed El Maria

Yara Essam Mohammed Ellkany

Abdelrahman Ibrahim Kamal Eldin

Mohamed Yasser Ahmad

Ali Tarek Ali Muhammed

Supervision

DR. Ali Ibrahim Siam

DEPARTMENT OF EMBEDEED NETWORK SYSTEMS TECHNOLOY

Graduation Project

2023

Smart Tour Guide

Submitted by:

- Mahmoud Hamza Abdella
- Abdelraouf Abdelmonem Abdelhakeem
- Ola Emad Naseem Zahran
- Alaa Taha Muhammed El Maria
- Yara Essam Mohammed Ellkany
- Abdelrahman Ibrahim Kamal Eldin
- Mohamed Yasser Ahmad
- Ali Tarek Ali Muhammed

Supervised by:

- DR. Ali Ibrahim Siam

Table of Contents

Preface

Tourism development is important.

Egypt is known among the world for the ancient Egyptian civilization, to which tourists and researchers come from all over the world to see its great ruins spread all over Egypt in museums, temples and tourist areas. Therefore, tourism is one of the most important sources of national income in Egypt.

More and more commercial and industrial sectors are stepping up their use of artificial intelligence (AI) to increase their operational capabilities, optimize processes and offer a better product/service to the customer. This is what AI is contributing to tourism, among many other things. Currently, AI has been incorporated into many areas of the travel and tourism industry, making lives easier for travelers around the globe.

The new technologies in the field of artificial intelligence, especially the field of natural language processing, as well as computer vision and machine learning, their effects at the present time and in the future will affect many different fields specially tourism to increase their operational capabilities Thus increasing the national income of the country.

Acknowledgement

It has been a great opportunity to gain lots of experience in a real project, followed by the knowledge of how to design and analyze our project. For that we must thank all the individuals who made it possible for students like us to complete that project. We would like to express our deepest gratitude to our faculty of artificial intelligence and our graduation project supervisor:

Dr. Ali Ibrahim Siam for her patience and guidance along the year. In addition, we would like to express our sincere appreciations to our graduation project coordinator Eng. Abdelmawla Yousef for her guidance, continuous encouragement, and great support.

We would like to thank President of Kafrelsheikh University and supervisor of the Faculty of Artificial Intelligence, Prof. Abdelrazek Desouki, and the Vice Dean of the Faculty of Artificial Intelligence, Prof. Tamer Medhat, for their efforts in the College to be the best.

And we would like to thank our college for the support, it provided us throughout the four years by providing all the available means for our production in the best and greatest way and extending us with science and knowledge to face the labor market after our graduation.

Finally, we would like to thank all the people who helped, supported, and encouraged us to successfully finish the graduation project.

Abstract

Today, tourism comes at the forefront of the sources of national income, and what serves tourism is to provide an exceptional experience for the tourist, and in response to this matter, this project was, which is a smart glasses equipped with a voice assistant capable of translating the hieroglyphic language, and just as the wonderful discovery of Champollion unveiled the secrets of the ancient Egyptian civilization, this The project will offer the tourist a translation of texts written in hieroglyphics without effort, in addition to the feature of identifying historical figures and presenting information about them. It is also equipped with the ability to identify tourist attractions in addition to translation between natural languages, which provides the tourist with what he needs without the need for a tour guide, which makes it wonderful for him and increases the demand for tourism.

Smart tour guide (STG) is a technology that combines augmented reality with tourism. It is a wearable device that can provide visitors with information about the places they are visiting, making their experience more immersive and informative. The smart glasses tour guide works by overlaying digital information onto the physical world, enhancing the visitor's view of their surroundings.

STG has a personal assistant capability that enables users to quickly access historical information by retrieving specifics about kings and queens from ancient Egypt. And incorporates landmark identification technologies to differentiate between various tourist and archaeological sites, aiding users in trip planning and itinerary optimization.

STG encourages tolerance and independence by allowing people who are deaf or dumb to read and comprehend Egyptian culture.

To achieve the objectives of the project, we will use the Sipeed Maixduino Kit RISC-V for AI, Deep Learning & IoT, which will be connected to servers over the network to enable us to implement artificial intelligence

and deep learning algorithms to recognize the old Egyptian language and provide instant translations.

Prerequisites

To understand this project you should be familiar with some approaches like, Python programming, micro python programing with maixpy, machine learning, deep learning, NLP, Speech Recognition, computer vision, server side programing and you should be aware with **Sipeed Maixduino Kit RISC-V.**

If you are a person who is not familiar with these approaches, we have given a simplified explanation of these areas in this book.

You can get project source code from GitHub with this link:

If you encounter any problems, you can contact the project teamwork on telegram through this link:

Chapter 1

Introduction

1.1 Introduction

With the great progress of technology in our time, it has become necessary to introduce technology in all areas of our lives from agriculture. Industry, commerce, and tourism.

The impact of technology on our daily life is much greater than we realize. It is developing and growing quickly. The way we access resources has altered as a result. It has also altered how we pick up new information. People these days frequently depend on technology for everything. We can instantaneously text someone whenever we need to get in touch with them. It used to move considerably more slowly because of letters and meetings. This is how technology has altered how we speak to one another. We are eventually testing the limits of technology and the ways in which it affects us as our needs and technical expectations continue to increase.

At this moment, it is challenging to envision a world without technology. The quality of life has changed significantly as a result. The way we behave, and function has altered because of technology permeating every part of our lives. Technology has improved every aspect of our life, from networking and healthcare to communication and transportation. The finest thing is that it always improves by enabling more sophisticated functions. For instance, facetime and instant messaging have come a long way from conventional voice calls. Not to mention that technology has helped during the pandemic. Video calling and posting messages on the internet have paved the road for people to contact with one another during times of global lockdown.

And through the new technologies in the field of artificial intelligence, especially the field of natural language processing, as well as computer vision and machine learning, their effects at the present time and in the future will affect many different fields.

The field of computer science known as "natural language processing" (NLP) is more particularly the field of "artificial intelligence" (AI) that is concerned with providing computers the capacity to comprehend written and spoken words in a manner like that of humans.

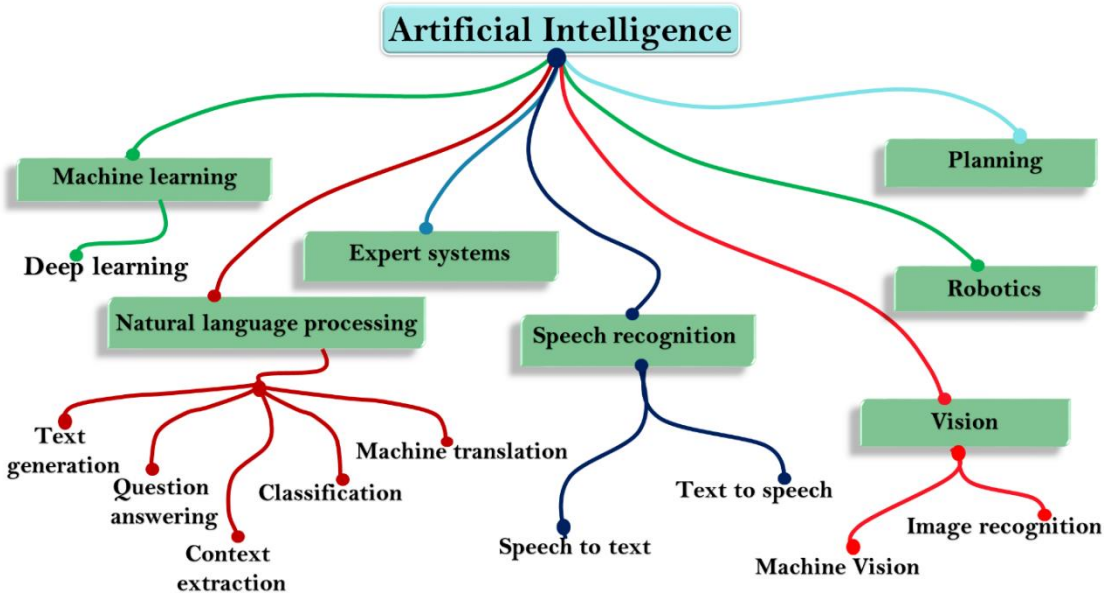


Fig 1.1: Artificial Intelligence branches

NLP blends statistical, machine learning, and deep learning models with computational linguistics—rule-based modelling of human language. With the use of these technologies, computers are now able to process human language in the form of text or audio data and fully "understand" what is being said or written, including the speaker's or writer's intentions and sentiment.

Computer programs that translate text between languages, reply to spoken requests, and quickly summarize vast amounts of text—even in real time—are all powered by NLP. You have probably used NLP in the form of voice-activated GPS devices, digital assistants, speech-to-text dictation programs, customer service chatbots, and other consumer conveniences. The use of NLP in corporate solutions, however, is expanding as a means of streamlining business operations, boosting worker productivity, and streamlining mission-critical business procedures.

Computer vision is a branch of artificial intelligence (AI) that enables computers and systems to extract useful information from digital photos,

videos, and other visual inputs and to execute actions or make recommendations based on that information. AI makes it possible for computers to think, while computer vision makes it possible for them to see, hear, and comprehend.

Humans have an advantage over computers when it comes to how eyesight works. The benefit of human sight is that it has had a lifetime to learn how to distinguish between things, determine their distance from the viewer, whether they are moving, and whether something is off about an image.

With cameras, data, and algorithms instead of retinas, optic nerves, and a visual cortex, computer vision teaches computers to execute similar tasks in a much less time. A system trained to inspect products or monitor a production asset can swiftly outperform humans since it can analyze thousands of products or processes per minute while spotting imperceptible flaws or problems.

It was necessary to use such fields to benefit from them in the fullest way, and within the integration between the different fields and activities, the choice fell on the field of tourism, which is an important source of national income for the country, and for all people to know the history of the ancient Pharaonic civilization.

Therefore, attention is paid to the comfort of the tourist first, then his knowledge of all the antiquities without the help of anyone, so that the knowledge is comprehensive, enjoyable, and entertaining always.

In Egypt, tourism is one of the most significant sources of income due to the annual dollar revenues it generates, the foreign exchange returns that allow it to contribute significantly to the GDP, and the ability to combat unemployment by employing a large portion of the Egyptian workforce. With its high number of visitors, Egypt is one of the most well-known tourist destinations in the world. It stands out for its wealth of tourist attractions of all kinds, the distribution of temples, museums, monuments, historical and artistic buildings, and vast gardens throughout its territory,

as well as its strong infrastructure built to support the tourism industry, which includes hotel rooms, villages, tourist resorts, tourism businesses, and airline offices.

Fig 1.2: Great Temple of Abu Simbel

And this is our solution, an easy and convenient way to serve tourists to see, read and know all the Egyptian antiquities that they visit so as not to waste their time, which increases tourism.

1.2 Purpose of the project

The Smart Tour Guide to design and develop a technological solution that enhances the experience of tourists visiting ancient Egyptian sites by providing them with accurate information, effective communication, and seamless planning. The Smart Tour Guide will be designed in the form of glasses, which will allow tourists to use it easily at any time and will be equipped with advanced hardware and software technologies to achieve its features.

Understanding the hieroglyphics and inscriptions on many Egyptian structures and artefacts requires the Smart Tour Guide's major function, which is its capacity to read and translate ancient Egyptian. The device also

contains a personal assistant that can look up details about any king or queen in ancient Egypt and provide visitors fast access to that knowledge.

Helping deaf and dumb people also to read Egyptian civilization and understand the people around them without anyone's help, and this will enable them to enjoy antiquities and civilization without feeling anything different from others, and it can also be used outside museums to learn to talk to anyone.

Knowing the antiquities by simply looking at the image to know the place in front of you using the Landmark detection feature so that it is easier to know the uses of the places, whether in videos that appear in museums or through pictures, or also by using them to move from museum to museum or from any other archaeological place.

1.3 Objectives

The main objective of this project is to design and develop a technological solution that enhances the experience of tourists visiting ancient Egyptian sites. The Smart Tour Guide will achieve this by providing visitors with accurate information, easy-to-use, Seamless Planning, the project aims to:

Easy-to-use

Our Smart Tour Guide is very easy to use as it can be controlled by voice commands, and Its design is like eyeglasses, enabling its user to benefit from its services effectively and quickly.

Accurate Information

Visitors will be able to decipher the hieroglyphics and inscriptions on Egyptian structures and artefacts thanks to the Smart Tour Guide. With proper interpretations and justifications, it will read and translate ancient Egyptian. The tablet will also have a personal assistant feature that allows users to look up information on any king or queen from ancient Egypt, giving users quick access to historical data. And there are other services like capturing photos, showing time, read QR-code, play music and other.

Seamless Planning

Visitors may organize their excursions to various tourist and archaeological sites in Egypt with the help of the Smart Tour Guide. In order to differentiate between distinct areas, the device will use landmark detection technology, delivering precise information and direction. The Smart Tour Guide will optimize itineraries and allow travelers to make the most of their stay in Egypt by providing a thorough planning tool.

Accessibility

The deaf and dumb community will benefit from the services provided by the Smart Tour Guide. They will be able to interact with others, read and comprehend Egyptian civilization, and take in the historic heritage without the need for outside aid.

1.4 Scope

The project will focus on the design and development of a Smart Tour Guide device for tourists visiting ancient Egyptian sites. The device will be equipped with advanced hardware and software technologies that enable it to recognize and translate ancient Egyptian language and provide instant translations. The project will also implement face recognition technology to provide accurate information about the ancient Egyptian figures and offers many other services.

1.5 Methodology

To achieve the objectives of the project, we will use the Sipeed Maixduino Kit RISC-V for AI, Deep Learning & IoT, which will be connected to servers over the network to enable us to implement artificial intelligence and deep learning algorithms to recognize the old Egyptian language and provide instant translations. The camera integrated into the device will

allow us to implement face recognition technology and provide accurate information about the ancient Egyptian figures.

We will also conduct extensive research on ancient Egyptian language and culture to ensure that the translations provided by the device are accurate and reliable. Additionally, we will conduct user testing to ensure that the device meets the needs and preferences of tourists visiting ancient Egyptian sites.

1.6 Deliverable

The deliverables of the project include:

- A fully functional Smart Tour Guide device that recognizes and translates ancient Egyptian language and provides instant translations.
- An integrated personal assistant that can look up details about any king or queen in ancient Egypt and provide visitors with fast access to that knowledge.
- Face recognition technology that provides accurate information about the ancient Egyptian figures.
- The usual services like QR-code reader, showing time and music player.
- Sign language recognition to help deaf and dump.
- Landmark detection algorithms to distinguish between Various tourist and archaeological sites.

1.7 Conclusion

The Smart Tour Guide project aims to enhance the experience of tourists visiting ancient Egyptian sites by providing them with accurate information, effective communication, and seamless planning. The project will use advanced hardware and software technologies to

achieve these features, and the Sipeed Maixduino Kit RISC-V for AI, Deep Learning & IoT will serve as the platform for the project.

Chapter 2

Hardware Specifications

2.1 Introduction

In this project we have made a smart tour guide. Let me tell you what the smart glasses tour guide is or what are its most important features and how to use it. Smart glasses tour guide is a technology that combines augmented reality with tourism. It is a wearable device that can provide visitors with information about the places they are visiting, making their experience more immersive and informative. The smart glasses tour guide works by overlaying digital information onto the physical world, enhancing the visitor's view of their surroundings. For example, the device can highlight points of interest, provide historical information about landmarks, or offer suggestions for nearby restaurants and activities. Additionally, the device can help reduce the workload of tour guides, as visitors can receive information without requiring constant attention from a human guide. One of the key advantages of the smart glasses tour guide is its convenience. Visitors can use the device hands-free, allowing them to fully immerse themselves in the experience without being distracted by a phone or a physical guidebook. Additionally, the device can be customized to the visitor's interests and preferences, providing a more personalized experience. And these glasses can also read and know the ancient Arabic language (hieroglyphics).

2.2 Hardware Components

This project contains some component such as:

1. **Display:** A small display that can be placed in front of the user's eyes to provide visual information.
2. **Camera:** A camera can capture visual information about the surroundings and enable features such as object recognition and augmented reality.
3. **Wi-Fi and Bluetooth:** Wi-Fi and Bluetooth can help the glasses to connect to the internet and other devices, such as smartphones or smartwatches.
4. **Audio components**: The glasses can include speakers and microphones to provide audio information and enable voice commands.
5. **Battery**: A rechargeable battery can power the device for an extended period.

6. **Software:** The software can integrate all the components and provide the user interface, information, and other features. The software can also incorporate machine learning algorithms for object recognition, language translation, and other intelligent features.

It contains the most important component **Sipeed Maixduino Kit RISC-V.**

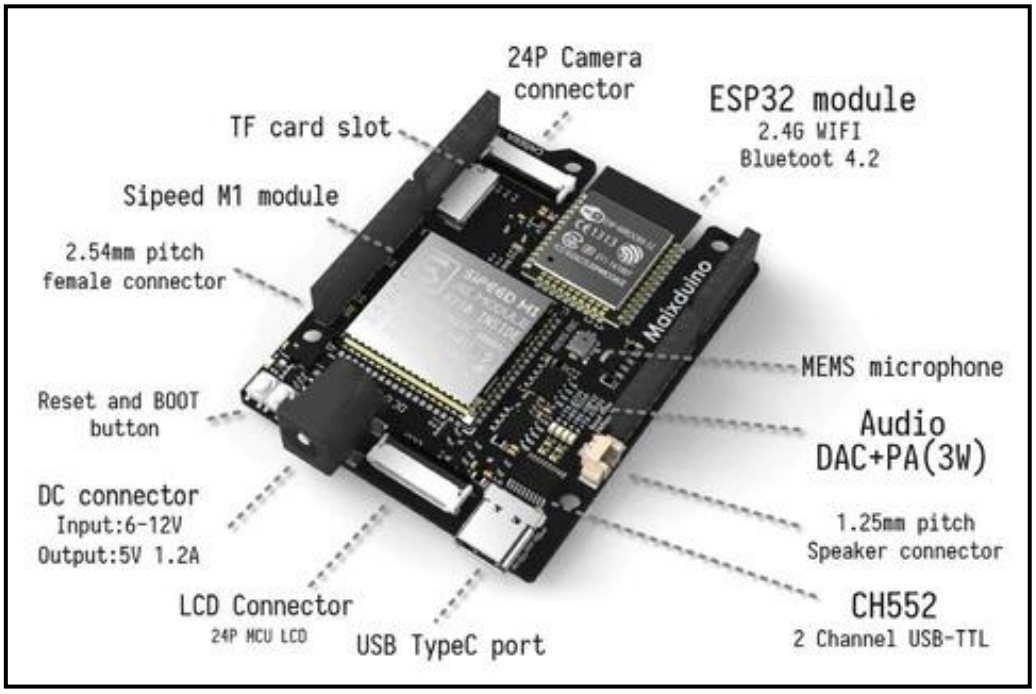


Fig 2.1: Sipeed Maixduino Kit RISC-V

2.2.1 Sipeed Maixduino Kit RISC-V

SIPEED MaixDuino is a development board compatible with Arduino based on MaixPy M1 module (main control: Kendryte K210). It integrates camera, TF card slot, user buttons, TFT display, MaixDuino expansion interface, etc., users can use MaixDuino to easily build a face recognition access control system, and reserve development and debugging interfaces, which can also be used as a powerful AI learning development board.

Artificial intelligence (AI), deep learning and machine learning embedded applications need a powerful processing board with special features.

Sipeed Maixduino AI development kit with integrated Micropython is an excellent start. Unlike the conventional C or C++ seen in most hardware boards, the Maixduino integrates Micropython to make development smooth and straightforward.

At the heart of the Sipeed Maixduino platform is the AI chip K210, a dual-core RISC-V with an FPU. It serves as a Dual-core processing with independent FPU, 64-bit CPU and 8M in-built SRAM. It supports multiplication, division and square root operation. Chip K210 can perform operations such as convolution, batch normalization, activation, and pooling. This module can realize face detection, voice, color and object recognition, MNIST handwritten digit recognition, feature map visualization, tiny yolov2 and so on.

Onboard functions:

Project	Description
CPU	Dual-core 64bit RISC-V / 400MHz* (double-precision FPU integration)
Memory	8MiB 64bit on-chip SRAM
Storage	16MiB Flash, support micro SDXC expansion storage (max 128GB)
Screen (package)	2.4 inch TFT, screen resolution: 320*240
Camera (package)	30W pixel GC0328 camera
DVP	Standard Camera DVP 24PIN interface
Power + USB	USB Type-C interface
ESP32	ESP32 SPI connection (ESP32 supports WIFI and Bluetooth), PAM8403A
DAC	I2C DAC
TF card slot	Multimedia resource expansion, support large-capacity storage

Table 1.1: Onboard functions

Maixduino with Smart Tour Guide (STG)

Sipeed Maixduino Kit RISC-V contains a camera, a processor, a memory up to one tera, a screen, Wi-Fi, and Bluetooth. It has an 8 megabyte of RAM, and its flash has a 16 megabyte. It has 2 buttons. One you use in your programming, and the other is the one that is reset, not reset You erase the code. but restart It. Working on power of 6 to 12 like an Arduino.

Sipeed Maixduino Kit RISC-V has many features like:

- Digital Video Port (DVP) interface support

- I2C, I2S, SPI and JTAG
- Field Programmable IO Array (FPIOA/IOMUX)
- Neural Network Processor (KPU)
- Audio processor (APU)
- Fast Fourier Transform (FFT) hardware accelerator
- AES and SHA256 algorithm hardware accelerators
- Machine vision and hearing
- Low power, better vision processing speed and accuracy
- KPU high performance Convolutional Neural Network (CNN) hardware accelerator
- FPC24P socket for DVP camera and 8-bit MCU LCD
- Power amplifier IC for use with speakers
- Microphone Array expansion board for sound localization, beam forming, speech recognition, etc.
- Onboard high-speed DAC
- On-board Wi-Fi, adopt ESP8285 chip.

What is the difference between MaixPy and MaixDuino?

Maixduino widget itself, But Maixpy is a firmware that operating system that manages the libraries that you will operate.

The memory of the chip is 6MiB general purpose memory + 2MiB AI dedicated memory, which is really very large compared to ordinary single-chip microcomputers. If AI function is not used, we can use the entire 8MiB memory. So many firmware versions were introduced and compiled, so the smaller the firmware, the more libraries it takes. In STG project we will use ***maixpy_v0.6.2_84_minimum_speech_with_ide_support*** firmware.

The firmware has been placed via kflash_gui software, where the board is selected, you choose the firmware you want to download, then you choose the board and then download the firmware on external SD card memory, when you come to make a boot, you take this first firmware and put it in the memory that is the RAM, and after that you go to run the file The one named *main.py*.

We will discuss how to load your firmware to the development board in chapter 4 in Kflash_gui section.

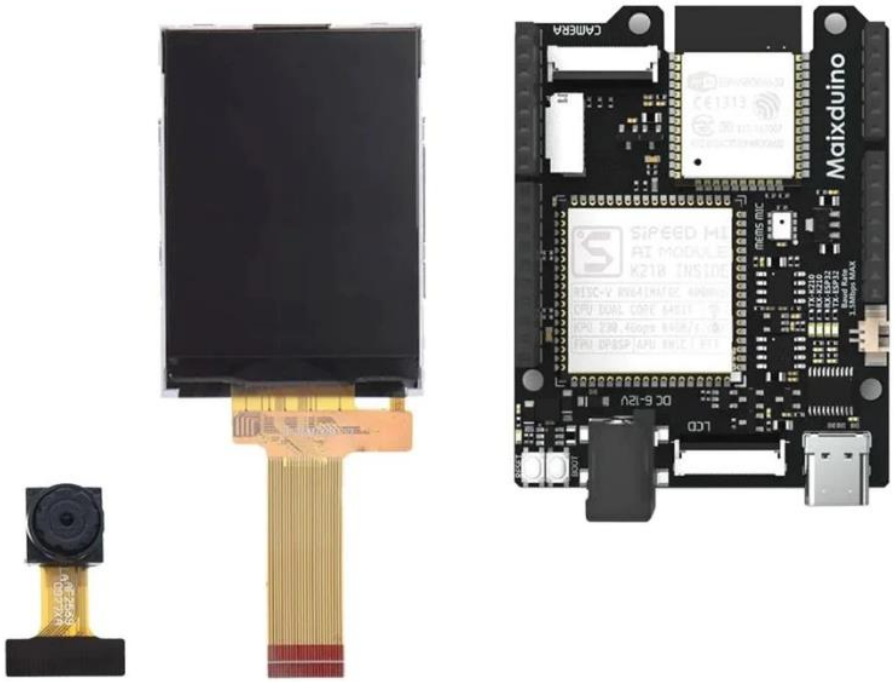


Fig 2.2: Sipeed Maixduino components

2.2.2 3D printing (A case for Maixduino)

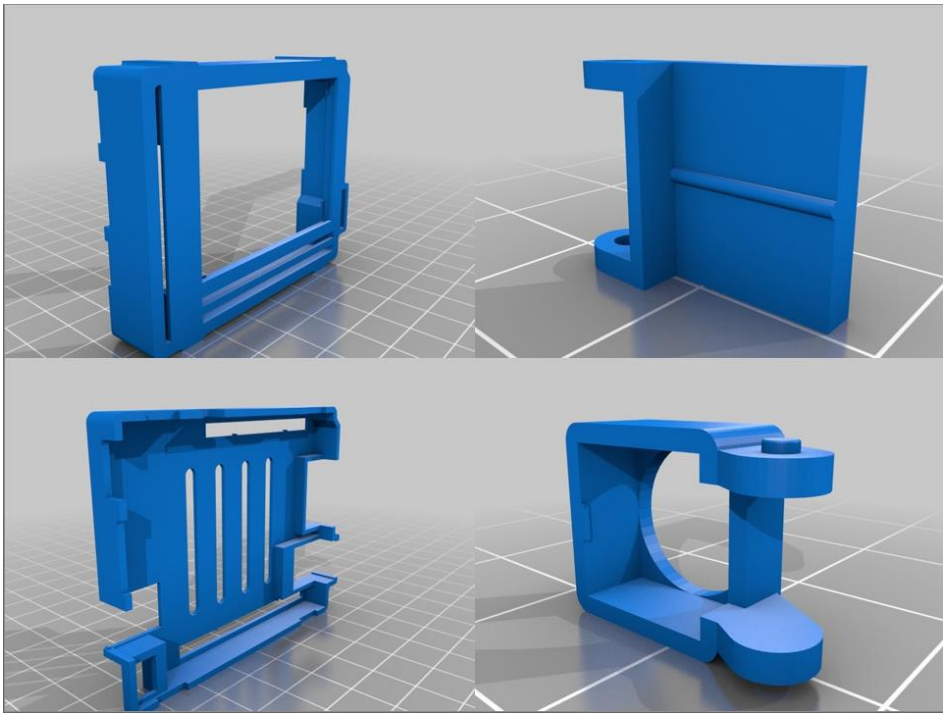


Fig 2.3: 3D printing (A case for Sipeed Maixduino)

3D printing or additive manufacturing is the construction of a three-dimensional object from a CAD model or a digital 3D model.

It can be done in a variety of processes in which material is deposited, joined, or solidified under computer control, with material being added together (such as plastics, liquids or powder grains being fused), typically layer by layer.

One of the key advantages of 3D printing is the ability to produce very complex shapes or geometries that would be otherwise infeasible to construct by hand, including hollow parts or parts with internal truss structures to reduce weight. Fused deposition modeling (FDM), which uses a continuous filament of a thermoplastic material, is the most common 3D printing process in use as of 2020.

In STG project we designed 9 CAD files, you can get STG CAD files from GitHub, See Prerequisites.

2.2.3 Audio components

MaixDuino board has Power amplifier IC for use with speakers, Microphone Array expansion board for sound localization, beam forming, speech recognition, etc.

In STG We used Audio “Plug Female 3Pin Vertical PCB” to make speaker portable.

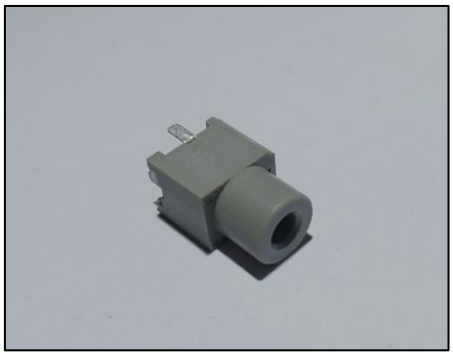


Fig 2.4: Audio Plug Female 3Pin Vertical PCB mount

Then we used headphone as a speaker.

Fig 2.5: Stereo headphone

2.2.4 Battery

For STG project we used has two batteries connected in series, meaning each battery 3.7 The two of them will work about 7.4 v, as MaixDuino need a power of 6 to 12.

Fig 2.6: Lipo Battery Cell 3.7V 1500mAh with Connector (40x30x10mm)

2.3 STG'S Hardware implementation

As shown in following figure that our smart tour guide is consists of two Lipo batteries, audio plug, one 3d printing, one stereo headphone, memory, screen, camera, and MaixDuino board.

Fig 2.7: Smart tour guide component

In the following figure, the batteries were connected in series to the MaixDuino each battery has a 3.7 V then put each components in its place.

Fig 2.8: Batteries connected to MaixDuino.

Upload your code on the memory then put the memory in the Maixduino. It can contain memory up to one tera.

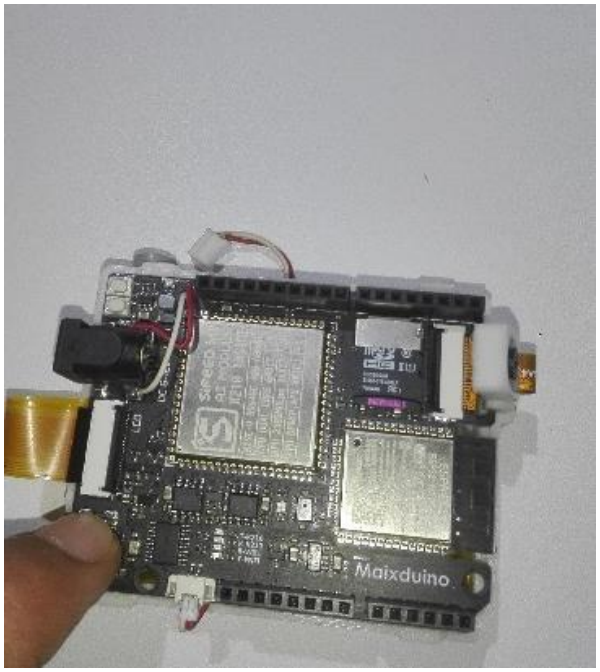


Fig 2.9: Memory 8 Mega in the MaixDuino

connect the camera to the Maixduino and connect a USB cable to test the camera and see the image quality.

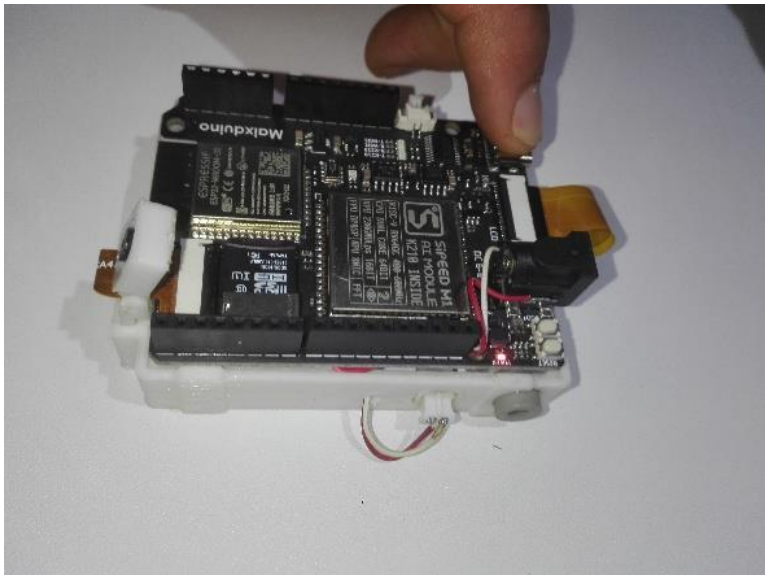


Fig 2.10: Connect Camera to Maixduino

Then we connected the screen to Maixduino and programmed it to show a specific text to test it as well.

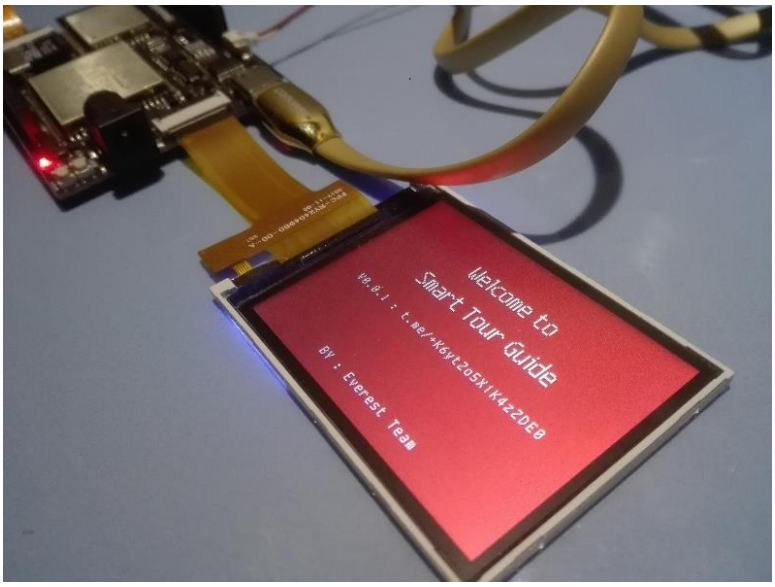


Fig 2.11: test MaixDuino screen

In the end, we put the Maixduino in 3D printing, so that we can then connect the headphones to it.

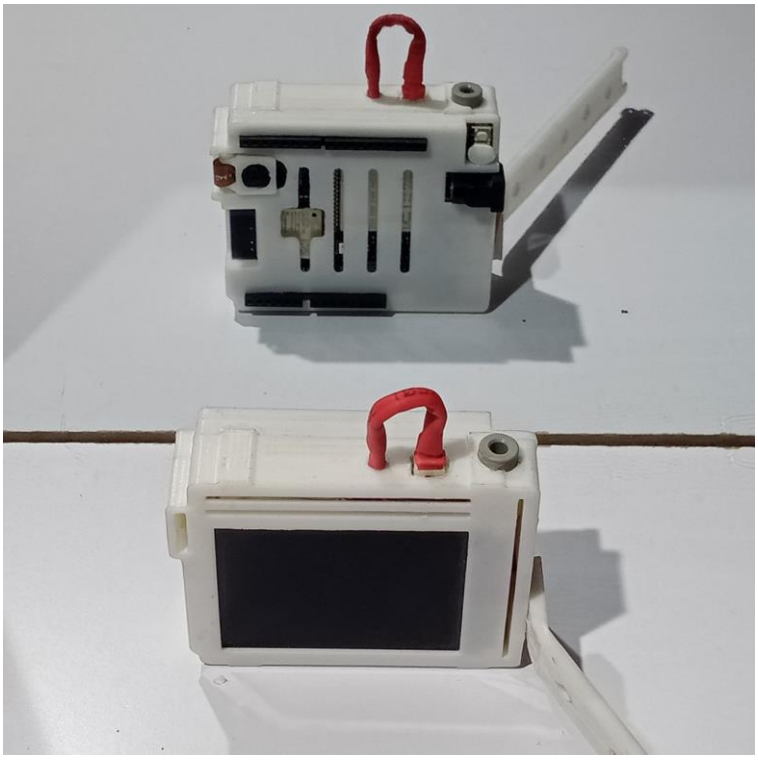


Fig 2.12: Maixduino in 3D printing

2.4 Conclusion

The Smart Tour Guide Glasses (STG) is a revolutionary device that can enhance the travel experience by providing users with personalized recommendations, language translation, navigation, and information on points of interest. With its advanced features and intuitive design, the Smart Tour Guide Glasses are an excellent choice for anyone looking to explore a new city or country.

Fig 2.13: Final design

Chapter 3

System Methodology

3.1 Introduction

In our project "Smart tourist guide system", we aim to exploit artificial intelligence and its tools in developing tourism in Egypt, improving the experience of tourists, ensuring their satisfaction in the best way, solving problems that may confront them, and helping to spread our great civilizations in the world.

Our system tries to simulate the job of a tour guide. It can help tourists understand ancient languages written on the walls of pharaonic temples and tombs and identify statues of kings and queens. The system is also supported with voice assistant to help tourists on their trip. Our system also supports the recognition of sign language to improve the experience of its speakers and facilitate communication with them.

In this chapter, we will discuss Smart tourist guide system in details.

3.2 System Design

The tourist can issue voice commands to the smart glasses, which will recognize and respond accordingly. If the command is related to the virtual assistant, the tourist can ask questions, and with the assistance of Google, the virtual assistant will provide audible responses. In the case of a translation command, the glasses will capture an image containing the text, which will be sent to the algorithm responsible for extracting and translating the text. The translation will also be provided audibly. If the command is related to facial recognition, the glasses will capture an image containing historical cherecters, which will be sent to the algorithm responsible for recognizing these cherecters. The glasses will then display the recognized figure accompanied by relevant information. For QR code commands, the glasses will capture an image containing the QR code, extract the information, and display the associated content. For capture commands, the glasses will simply capture and save the images. If the

command is related to music, the glasses will play the requested music, while for time-related commands, the glasses will display the current time to the tourist.

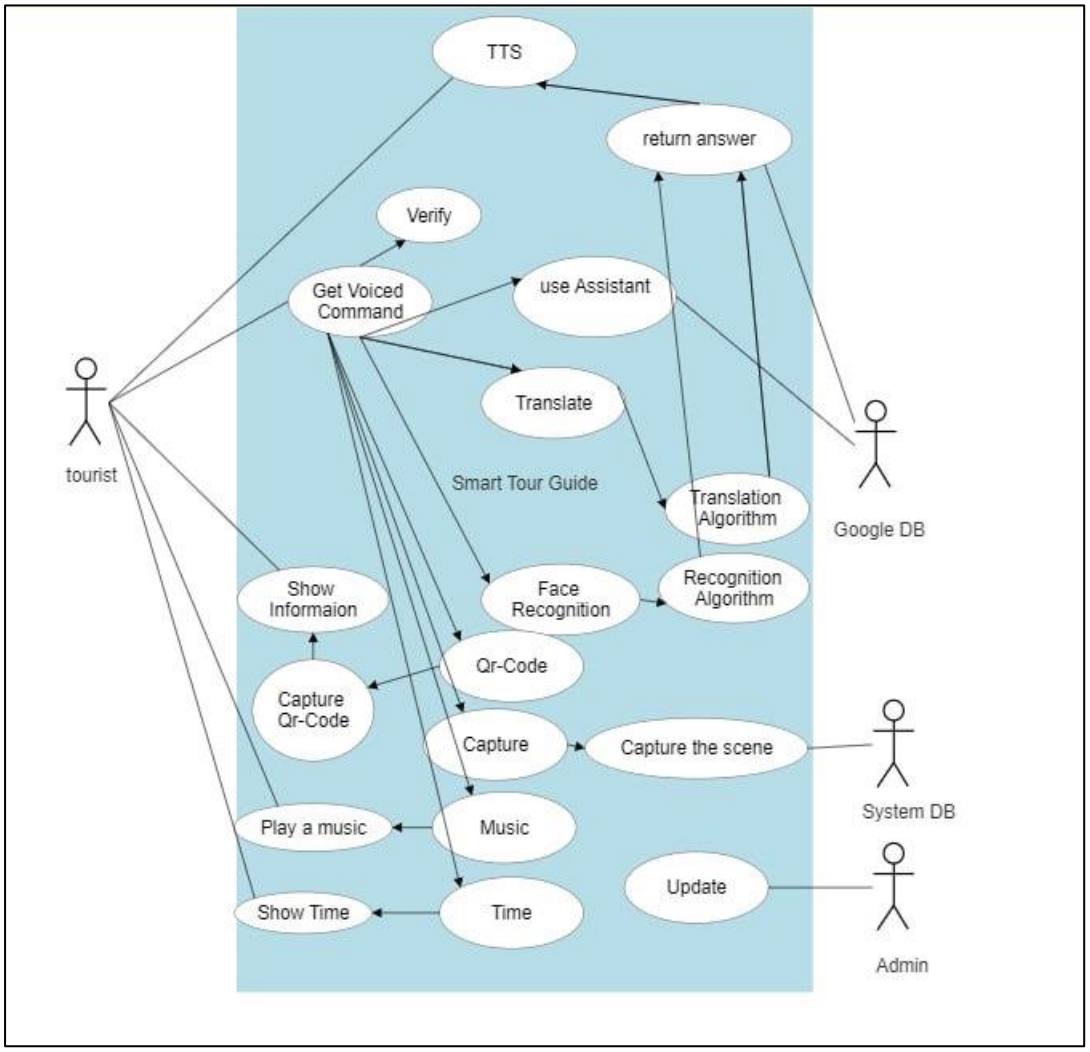


Fig 3.1: STG system use case

3.3 Main functions in STG System

Here are the main functions of the system in details:

3.3.1 Translate ancient languages.

Introduction to hieroglyphics

hieroglyph, characters used in a system of pictorial writing, particularly that form used on ancient Egyptian monuments. Hieroglyphic symbols may represent the objects that they depict but usually stand for a particular

sounds or groups of sounds. Hieroglyph, meaning “sacred carving,” is a Greek translation of the Egyptian phrase “the god’s words,” These signs were arranged in rows or columns and were read from right to left or left to right, depending on the direction in which the hieroglyphs faced.

Our project provides a wonderful approach to translating that language, enabling tourists to comprehend the engravings and texts written in the hieroglyphic language.

Fig 3.2: symbols of hieroglyphs

Challenges with translating hieroglyphs

The first challenge we faced is that Hieroglyphic writing consists of about 500 known signs, in addition to its multiple and difficult grammar. Hieroglyphs are difficult to learn, and learning takes a very long time. our solution is to work on certain signs and grammar. In hieroglyphic murals, the names, their prefix and their suffix are written in an oval shape called cartouche. This cartouche contains some signs construct the word. In this project we translate the names found in hieroglyphic murals, their prefix and their suffix with some level of generalization in translating.

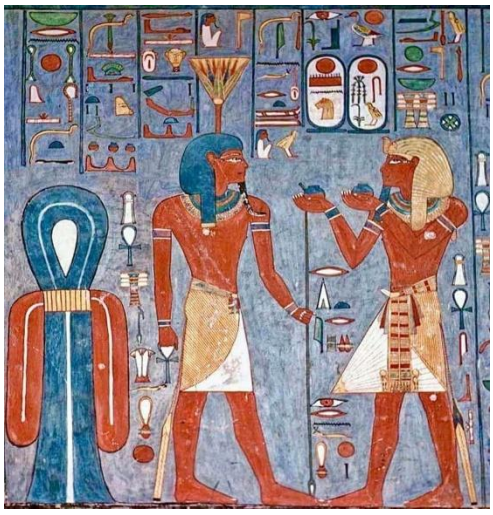


Fig 3.3: hieroglyphic mural and cartouche

Fig 3.4: cartouche (Ramses name)

The second challenge we faced is to collect and prepare hieroglyphs data. The idea of this project has not been implemented before, so this project is the first of its kind. There wasn't any prepared data for hieroglyphs.

Data Gathering:

For the purpose of this thesis, we gathered images of hieroglyphic murals, since there was not any existing dataset concerning our aims available, the data collection was created manually and consumed long time.

Data Preprocessing:

As we use deep neural network (shown later in this part), there is a need for large amount of data as possible. We used annotation to increase the data up to 15,000 images. Then we increase image clarity to ensure that signs in images are clear when applying the model.

How translate hieroglyphs system works

The hieroglyphics consist of signs and symbols. And our task is to detect and recognize this signs and symbols. This task is considered an object detection and recognition task.

YOLO algorithm is one of the best solutions for object detection task. So, we used YOLOv5 in this project.

The system consists of two main modules:

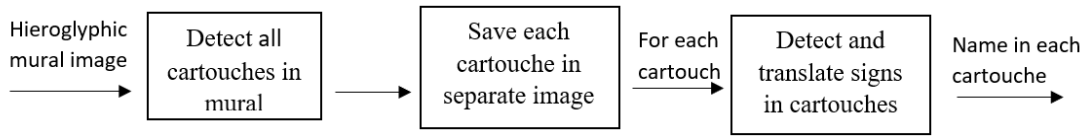


Fig 3.5: steps of translating names in hieroglyphics

I. Detect cartouche in the hieroglyphic mural:

As we explained before, our task is to translate names and their prefix and suffix which may be king's birth name or king's throne name in hieroglyphic mural. Names, prefix and suffix are in an oval shape called cartouche in the hieroglyphic mural. All cartouches in the same mural are related to the same king. However, each cartouche contains different words. So, we translate all cartouches in the mural. So first the model

detects all cartouches in the mural and calculates the area for each cartouche. Converts each cartouche to a separate image. The model saves these images to be the input for second module. It is an object detection task. We used YOLOv5 to detect cartouche in the hieroglyphic mural.

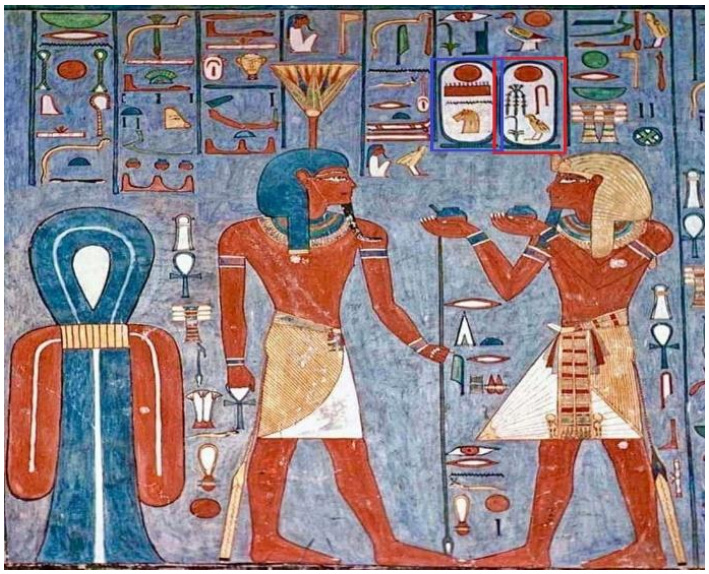


Fig 3.6: detect cartouches in mural.

II. Detect and Translate signs in cartouche.

After we detect cartouches in mural and save them as new images. The second step in to detect and recognize signs in the new image. Cartouches contain names of the kings will be lliterally translated. Other cartouches contain suffix such as king's birth name or king's throne name could be either translated literal translation or general translation.

Words in hieroglyphs are written in complicated style from top to bottom and from right to left.

The steps of translating name as flow:

1. detecting the signs in the cartouche using YOLOv5.

Fig 3.7: detect signs in cartouche.

2. then arrange signs to construct the name correctly:
   - First the model starts to arrange from top to bottom Based on Y coordinates as the symbol with the minimum vertical coordinate value comes first.
   - If there are two signs that intersect vertically, it looks for x-coordinates for two signs. a check is performed on their horizontal coordinates. If the minimum value of the horizontal coordinate of one symbol is greater than or equal to the maximum value of the horizontal coordinate of the other symbol, the symbol with the larger horizontal coordinate takes precedence and is placed before the other symbol.
   - The process continues until all symbols are sorted based on these criteria.

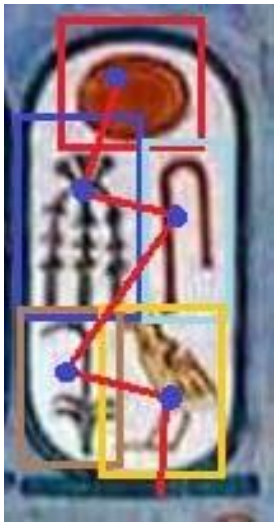


Fig 3.8: arrange signs in cartouche.

This arrangement algorithm successfully works for some cartouches. But we found that there are other cartouches that do not use this algorithm. So,

we build simple model to deal with them. Finally, the model output the name in cartouche in English.

3.3.2 Identify statues of kings and queens

The second feature in our system is to identify statues of kings and queens of ancient Egypt in tourist attractions.

One of the most important ancient Egyptian monuments found in most archaeological sites are the statues of the kings and queens of Egypt, which were carved to perpetuate their achievements. This service helps tourists to identify the king or queen, the owner of the statue and then tourists can search about them and know about pharaohs and their achievements.

In term of artificial intelligence, face recognition refers to the technology capable of identifying or verifying a subject through an image, video, or any audiovisual element of his face. Face recognition now is one of the most important technologies in the world. It is widely used in different fields. Face recognition models commonly encompass a range of tasks, including recognition, identification, verification, categorization, and the analysis of facial expression. It is used in security, social media, face attendance in schools or universities, the attendance of employees in a firm or an entity…etc.

In this project, we use a Deep Model for face detection and recognition to recognize the faces in the ancient statues of the kings provided in the history background.

For this service, there are two versions:

- The first one is one we use in this project. it is written in Micropython, it is more suitable for Maixduino board. It is built based on YOLO algorithm. It is very fast and accurate but it is not suitable for large number of faces.

- The second one is one which is proposed in python and is suitable with boards that have higher resources like: Raspberry pi or computers. It uses python libraries such as: OpenCV and face recognition. It can recognize more than 25 people of ancient kings and queen.

3.3.2.1 Applied version in Micropython.

Here are the steps of face recognition:

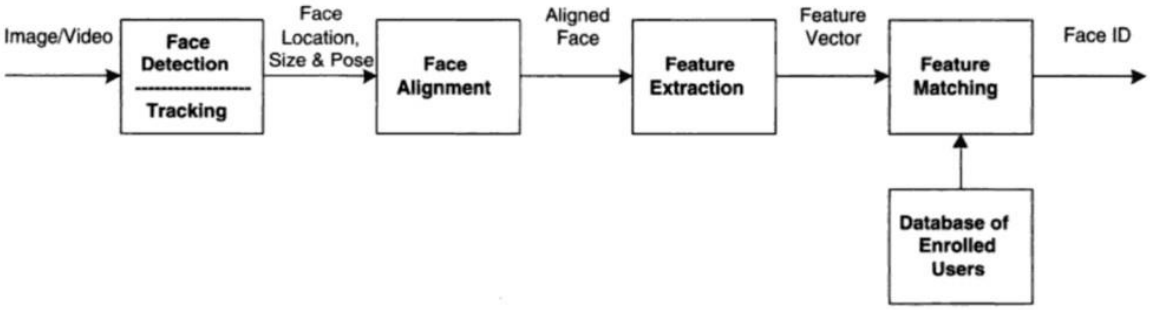


Fig 3.9: steps of the model

Step 1: Face detection

Detection is the process of finding a face in an image. The camera detects and locates the image of a face in statues, either alone or in a crowd. In our live tour guide glasses, the system can detect the face in multiple frames.

Machines use computer vision to identify people, places, and things in images with accuracy at or above human levels and with much greater speed and efficiency.

Most modern systems use an image-based approach and machine learning. They include neural networks that are trained to detect faces by looking through large numbers of pictures and making decisions about whether a part of an image is also a part of a human face. This is the cutting edge of computer vision, offering better real-time performance and effectiveness.

In our project, we used **Yolov2** to detect faces which is based on **deep convolutional neural network.**

Step 2: Face landmarks detection.

That involves detecting and localizing specific points or landmarks on a face, such as the eyes, nose, mouth, and chin to use them in face recognition.

In our project, we used pretrained model specific for Maixduino board to detect landmarks in faces.

Step 3: Extract facial features.

This module is responsible for composing a feature vector that is well enough to represent the face image. Its goal is to extract the relevant data from the captured sample. Template generation is the result of the feature extraction process. A template is a reduced set of data that represents the unique features of an enrollee's face consisting of weights for each image in the database.

In our project, we used pretrained model specific for Maixduino board to extract facial features.

Step 4: Recognize face image.

The Last step is to compare the template generated in step three with those in a database of known faces. The system reads a sample and compares that sample against every record or template in the database, this process returns a match or a candidate that is closest to the generated template from the database. Closest match is found by using the Euclidean distance which finds the minimum difference between the weights of the input image and the set of weights of all images in the database.

This version of face recognition project applied in MicroPython and is especially most suitable for Maixduino board.

Fig 3.10: face recognition on STG

3.3.2.2 Sub project: proposed version in python

At beginning, this version of statues face recognition project was the main version. This version is applied in Python language and uses Python libraries such as: OpenCV and face recognition library which is not supported in Maixduino board. So, this project is suitable when using boards like Raspberry pi or to run on computers.

In this version we mainly use face recognition module in python to detect and identify statues of kings and queens of ancient Egypt. It is a deep learning model. This model first detects faces in image or video, detect landmarks on faces and extract the facial features of each face as an array then compare this array with training data and choose the closest data point as the best match for this face.

The steps in this project:

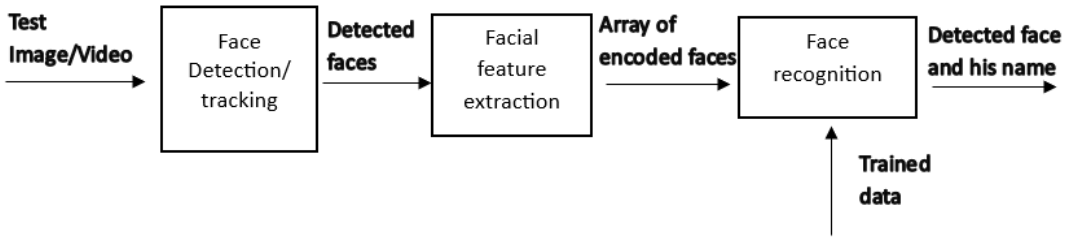


Fig 3.11: steps of the model

I. Gathering data:

For this project, we gathered images regarding the 20 different Egyptian kings and queens, since there was not any existing dataset concerning our aims available, the data collection was created by manually downloading images from the web. The data should be split into Training and Testing.

II. Preparing training data for model:

In this step, we change every face in the training set for an encoded array to identify the faces features. Based on this encoding, we can measure the similarity between two face images — that can tell us if they belong to the same person.

III. Apply the model:

The testing data is fed into the system either it is an image or a real-time video. The face recognition module first detects all faces in image or frame and tracks them. As flow:

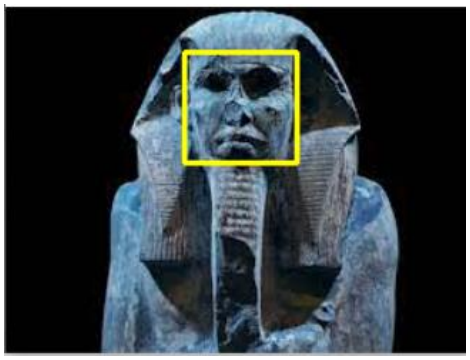


Fig 3.12: detect face with face recognition module

Then it encodes each face in an array carries the features of the face. Finally, for each face it compares this encoded face array with all training data and measures the similarity then the most similar datapoint is chosen as the predicated class. As flow:

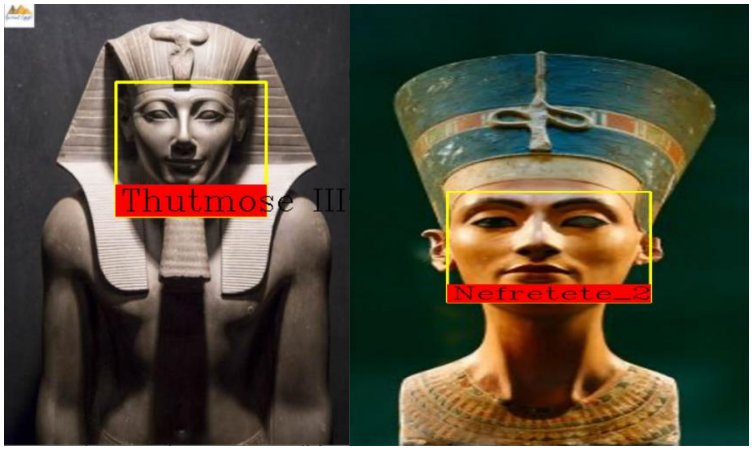


Fig 3.13: face recognition with face recognition module

3.3.3 Voice Assistant

This one of the most exciting features in our system. Voice assistant help tourists in their trips and facilities the communication between tourist and the glasses. It helps users to use the glasses via voice-commands and get response as speech instead of text. This voice assistant named Alex as default and each user can change the name as he likes. It can help tourist in many things such as: tell the time and the weather, search about something on google or Wikipedia, open videos on YouTube, play music and can translate languages to help people translate the result of the hieroglyphics language translation which is in English to the user language and help tourists from different countries to communicate.

This service to be fully implemented requires higher processing resources than Maixduino board. We implemented two versions of this service to cover the two cases:

- The first one is one we use in this project. It is customized Voice assistant. It is more suitable for Maixduino board. It uses Voice assistant to the most important function: "searching using Voice assistant" which helps tourists to know more about ancient Egyptian civilization, kings and queens, and landmarks they see in their trip or search about anything else.

- The second one is one which is proposed as a full Voice assistant. It is suitable with boards that have higher resources like: Raspberry pi or computers. It can do all functions related to Voice assistant such

as: know time, date, and the weather. Searching about anu thing, open YouTube or Google ...etc.

How voice assistant work?

In all cases, there are main steps Voice assistant follows to work:

1. First it can listen to human voice, save it and convert it to text using Speech recognition Module.

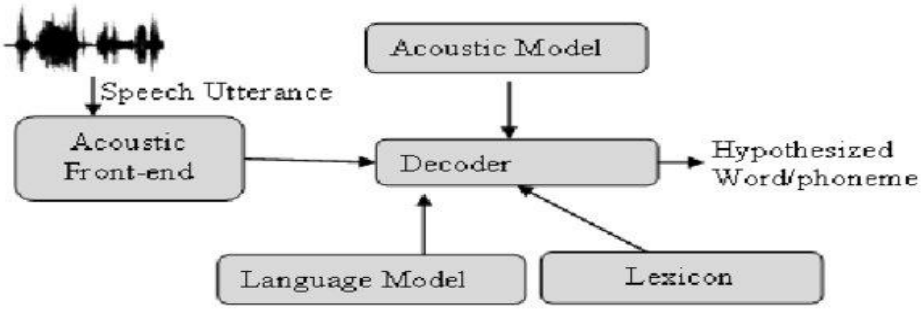


Fig 3.14: speech to text process

2. Based on text generated, voice assistant defines which function to do and do it.

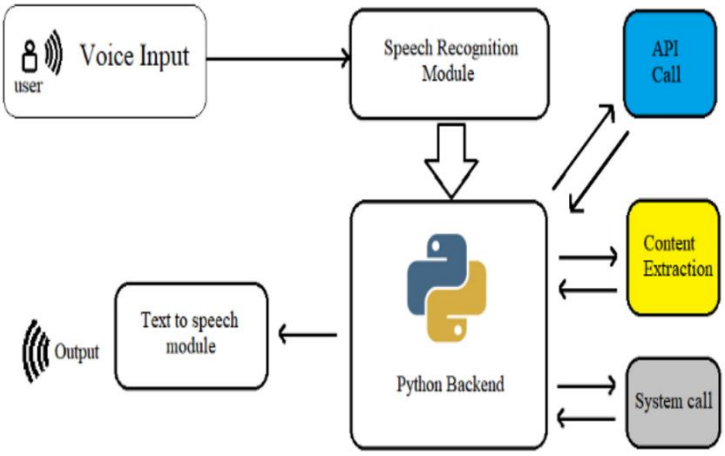


Fig 3.15: steps of voice assistant

3. The output generated as text Voice assistant converts it to speech again using TTS Module and out it to human.

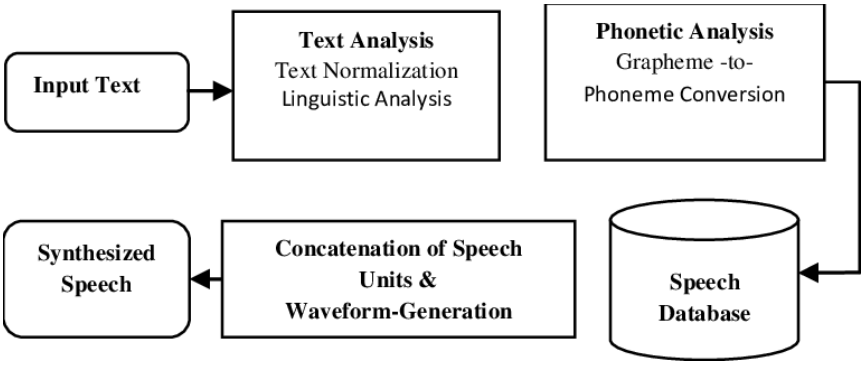


Fig 3.16: text to speech process

3.3.3.1 Applied version of Voice assistant:

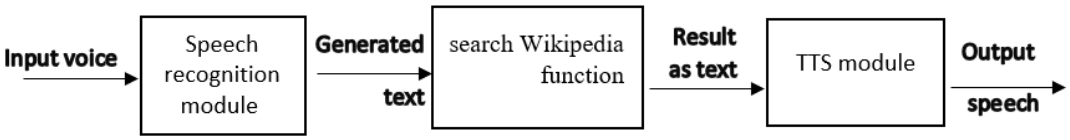


Fig 3.17: steps of customized Voice assistant

1. First, user opens Voice assistant service using "assistant" voice-command.
2. The user asks Voice assistant what he wants as a speech, this speech is translated into text by using speech recognition module.
3. Voice assistant call search_wikipedia function to search about what user want.
4. The result generated as text Voice assistant convert it to speech using TTS module and output the speech to the user.

3.3.3.2 Sub project: proposed version of Voice assistant:

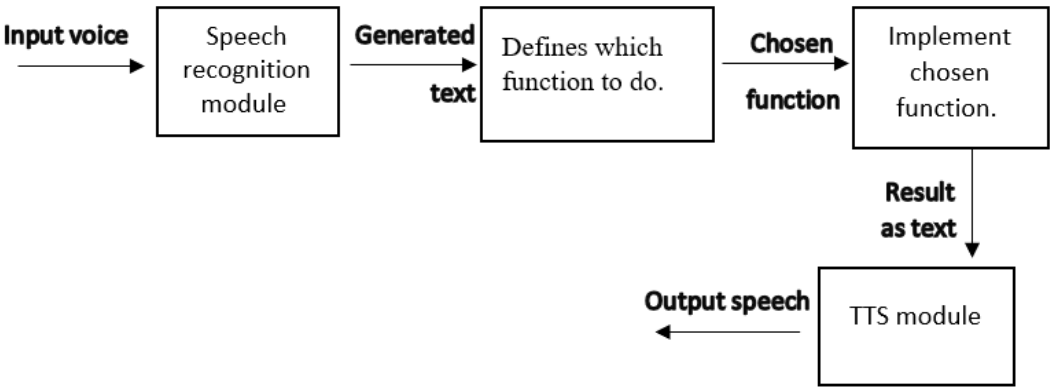


Fig 3.18: steps of full Voice assistant

1) Once user open Voice assistant service, it first wishes him and ask him what he want.
2) The user asks Voice assistant what he wants as a speech, this speech is translated into text by using speech recognition module.

3) From generated text, voice assistant defines what function the user wants and activate this function. Voice assistant can do multiple functions such as:

   - It can do Wikipedia searches for user.
   - It can open websites like Google, YouTube.
   - It can translate for user.
   - It can play music for user.
   - It can play videos for user.
   - It can tell user the time, date, and weather.
   - It can entertain user through jokes.
   - It can help user to make notes.

4) After voice assistant activate the desired function and get response as text then it converts text to speech by using TTS module and output the speech to the user.

Ex: tourist can ask Voice assistant to search about Ramesses II, Voice assistant activate the function responsible of searching on Wikipedia and get the first 4 sentence as result, finally TTS module convert the text to speech and output the speech to the user.

Using voice assistant in translating natural languages

The most important usage for Voice assistant is its ability to translate natural languages. Tourists and researchers from all over the world come to enjoy the great ancient civilization of Egypt. They certainly speak a variety of different languages. So, this powerful translation tool is mainly important to help tourists from different countries to communicate. Also, the result of the hieroglyphics language translation is in English, so this feature can be used for translating the output of the hieroglyphics language translation in English into any other language tourist want. This service can translate all the common languages in the world (more than 200 natural languages).

How voice assistant work:

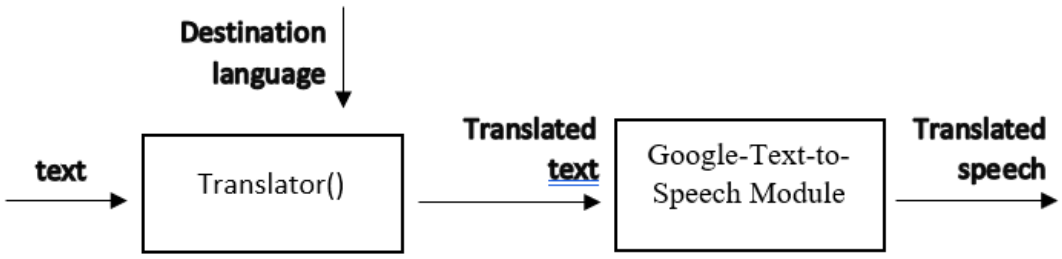


Fig 3.19: steps of translation

1. First Voice assistant service should be open.
2. From Voice assistant service user should call the translating function.
3. For translation, first user chooses the language to which translate.
4. The service uses translator function from Googletrans module to translate the input text.
5. Using Google-Text-to-Speech i.e., gTTS() method to speak the translated text into the destination language.

3.3.4 Landmark and Monuments recognition

The fourth feature in our system is Landmark Detection. This feature enables tourists to identify the monuments and thus they can learn more information about these monuments and their history by searching with the voice assistant.

Monument recognition is a challenging problem in the domain of image classification due to huge variations in the architecture of different monuments. Different orientations of the structure play an important role in the recognition of the monuments in their images. In this system, we propose an approach for classification of various monuments based on the features of the monument images. The model is trained on representations of different historical Egyptian monuments, obtained from cropped images, which exhibit geographic and cultural diversity. Experiments have been carried out on the manually acquired dataset that is composed of images of different monuments where each monument has images from different angular views. The experiments show the performance of the model when it is trained on representations of cropped images of the various monuments. The overall accuracy achieved, and confidence is high using YOLOv5 algorithm, for a total of 10 different monuments that have been considered in the dataset for classification.

At this point, it is considerable to mention that the recognition should allow the possibility of the various lighting conditions, different points of view as well as the potential partial occlusions. The fact that the geometrical shape of the desired objects is often like many other objects and a minor change of the angle may provide totally different information, renders this task challenging.

Landmark detection is still object detection task. So, using YOLO algorithm is one of the best solutions for this task.

The steps for this task as flow:

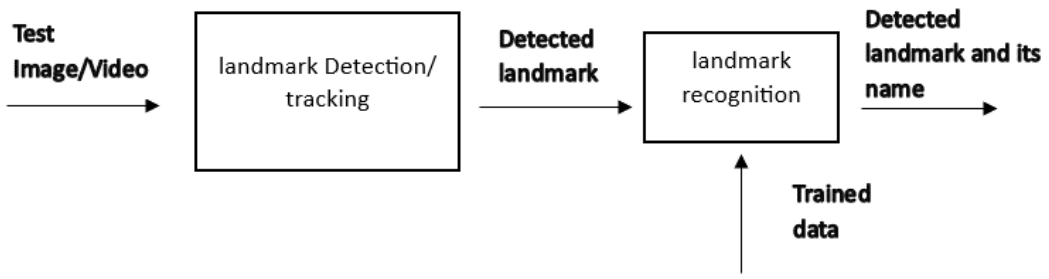


Fig 3.20: steps of the LM model

1. Preparing the dataset:

Probably the most time-consuming part of the experiment consisted of the data gathering and the pre-processing. An essential and inextricable task for every training model in machine learning or deep learning is the collection of the required dataset (collection of data).

- Data Gathering:

For the purpose of this thesis, we gathered images regarding the 10 different Egyptian Monuments, since there was not any existing dataset concerning our aims available, the data collection was created by manually downloading images from the web, in combination with personal photographs taken for the respective monuments/landmarks. As lighting conditions and different angles of a monument are two factors that directly affect the coloring and possibly the shape of a monument, we try to maintain a variety in our images. With this strategy, it is ensured a greater chance for a correct and accurate prediction, even under various circumstances. images data are divided into 10 classes.

Examples of classes are presented here :

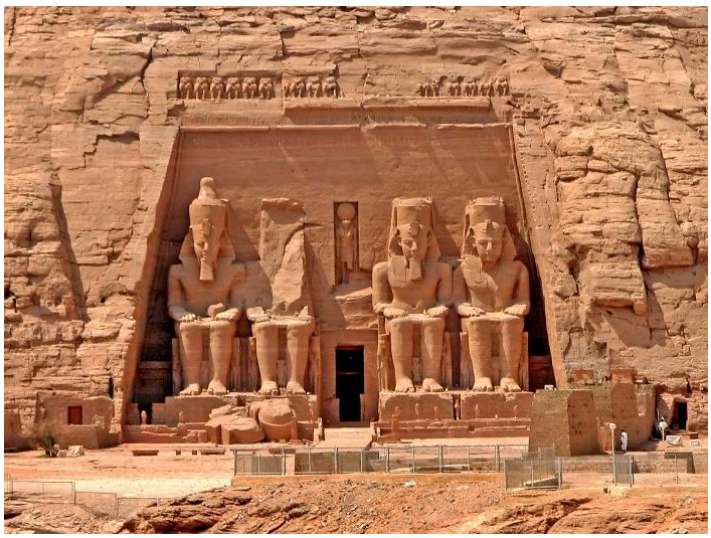


Figure 3.21: abo simple temple

Figure 3.22: Luxor temple

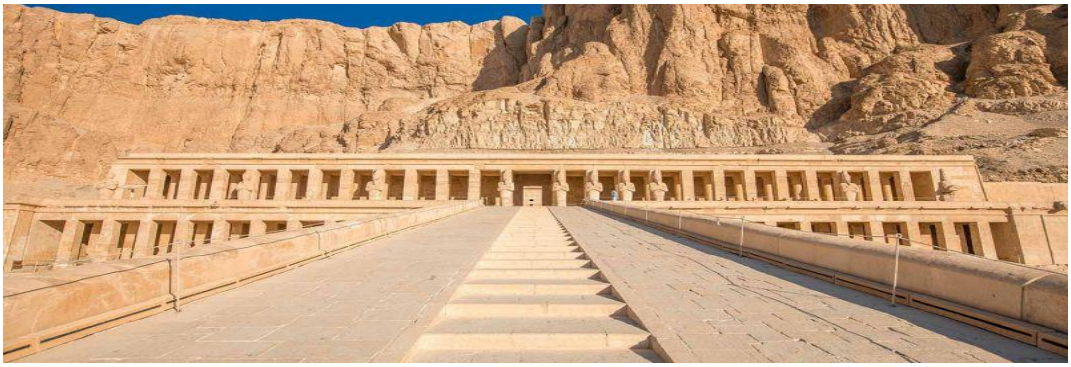


Figure 3.23: Hatshepsut temple

- Checking the quality

First, it is vital to maintain only the images that satisfy a certain resolution. Images with a very low resolution are not useful for our purpose, as the information gained from them is minor and consequently, the overall performance is decreasing.
In order to avoid any errors regarding the name of the images, which often pose a great length and contain punctuation marks and spaces, we renamed each image providing a meaningful to us name. Aside from this, we have also the potential to easier pinpoint and process the desirable image.

- Annotating the images

A time-consuming but fundamental part of the pre-processing is the annotation of the images. To annotate the images, we use a tool named LabelImg, available for several platforms, enabling us an easy draw of the desired bounding box along with the annotation for each box, as seen in the following figure. LabelImg also offers us

the opportunity for saving the annotation with either YOLO or PascalVOC format. And we can make annotation with Roboflow

Roboflow: is building gems of tooling to make image annotation easy. This feature, polygon bounding boxes, and label assist.

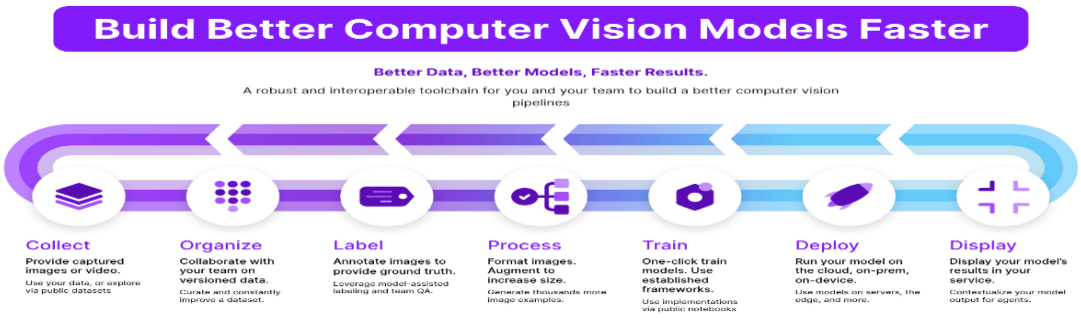


Fig 3.24: data annotation

object detection. After the confirmation for the bounding box and the annotation, the program automatically generates a (XML) file with the same name as the image. For each image, we can observe that the (XML) file, besides the filename and the size of the image also contains the label name in association with the coordinates of the drawn bounding box/boxes.

2. Build the model .

After preparing data, now we can use YOLO algorithm to build our model. YOLO stands for You Only Look Once. YOLO algorithm divides an image into the grid system and in that each grid detects objects within itself. Now the latest version of YOLO is V5 which is launched by Ultralytics. This YOLO V5 algorithm is the best of all object detection algorithms available so far. It is simple, easier, and faster. It detects objects with high accuracy.

As YOLO algorithm works, the algorithm applies a single neural network to the full image. This network divides the image into regions and predicates bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicated probabilities. YOLO algorithm then identifies the appropriate class for each bounding box based on these probabilities.

As a result, the model predicates the class for each monument as flow:

Fig 3.25: model prediction

3.3.5 Sign Language Detection

The last feature in our system is sign language detection. This feature has been added to help people speak with sign language to communicate with the others easily. It helps tourists to have a better experience. This can be very helpful for the deaf and dumb people in communicating with others as knowing sign language is not something that is common to all.

This feature is divided into two main modules as flow:

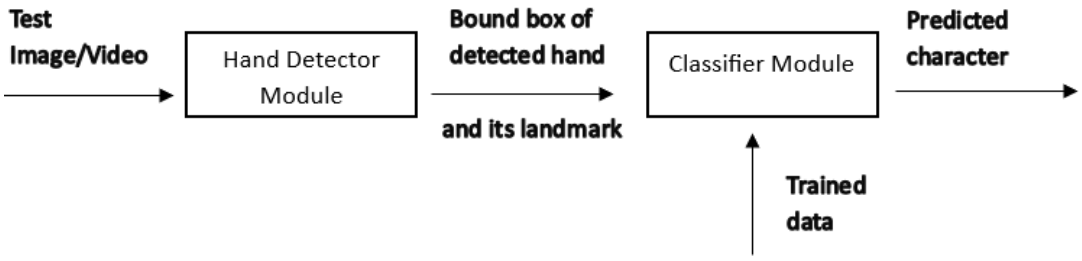


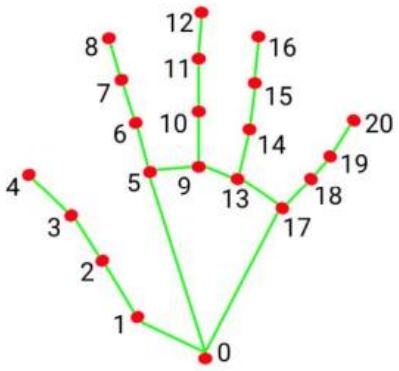
Fig 2.25: steps of the model

I. Hand Detector Module:

is a key component of a hand tracking module that is responsible for detecting the presence of hands in an image or video frame. The hand detector is usually based on computer vision algorithms or machine learning models that are trained to identify the shape, color, texture, and other features of human hands. Once the hand detector has identified the

position and orientation of the hands in the image or video frame, the hand tracking module can use this information to track the movement of the hands over time. This involves estimating the 3D position and orientation of the hands relative to the camera or sensor and predicting the next position of the hands based on their past movement. Hand detectors can vary in their accuracy and robustness, depending on the quality of the training data, the complexity of the algorithms or models used, and the environmental conditions in which the hand tracking module is used. Some hand detectors may be optimized for specific types of hands or hand poses, while others may be more general purpose and able to detect a wide range of hand shapes and movements. Overall, the quality and performance of a hand tracking module depend on the effectiveness of its hand detector, as well as the other components of the module, such as the tracking algorithm, the user interface, and the hardware platform.

The MediaPipe Hand Landmarker task lets you detect the landmarks of the hands in an image. You can use this Task to localize key points of the hands and render visual effects over the hands. This task operates on image data with a machine learning (ML) model as static data or a continuous stream and outputs hand landmarks in image coordinates, hand landmarks in world coordinates and handedness (left/right hand) of multiple detected hands.



0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

Fig 2.26: 21 landmarks points on hand

The hand landmark model bundle detects the key points localization of 21 hand-knuckle coordinates within the detected hand regions. The model was trained on approximately 30K real-world images, as well as several rendered synthetic hand models imposed over various backgrounds. The hand landmarker model bundle contains a palm detection model and a hand landmarks detection model. The Palm detection model locates hands

within the input image, and the hand landmarks detection model identifies specific hand landmarks on the cropped hand image defined by the palm detection model. Since running the palm detection model is time consuming, when in video or live stream running mode, Hand Landmarker uses the bounding box defined by the hand landmarks model in one frame to localize the region of hands for subsequent frames. Hand Landmarker only re-triggers the palm detection model if the hand landmarks model no longer identifies the presence of hands or fails to track the hands within the frame. This reduces the number of times Hand Landmarker triggers the palm detection model.

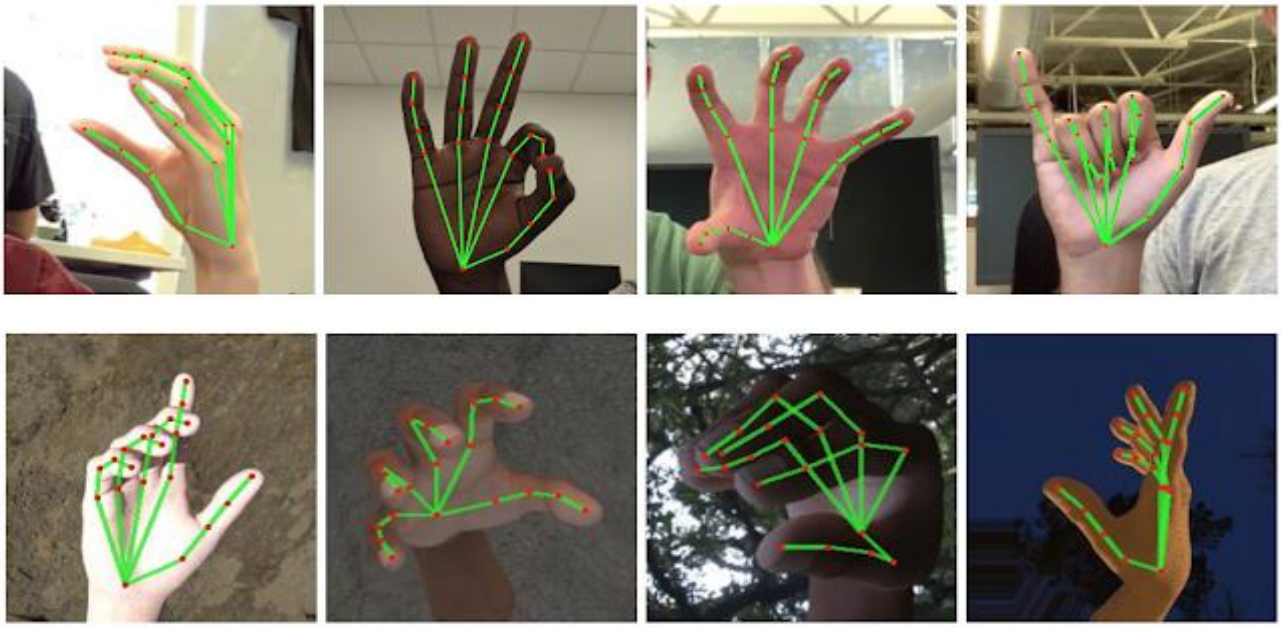


Fig 2.27: Sample of Hand detection results

II. Classification Module:

also known as a classifier, is a type of machine learning model that is trained to assign input data to one of several predefined categories or classes. The classifier takes a set of features as input and outputs a predicted class label based on its learned knowledge from the training data. Classification modules are commonly used in a wide range of applications, such as image recognition, speech recognition, natural language processing, fraud detection, and spam filtering.

In sign language, each hand sign refers to a certain character. And the collection of signs can construct a certain word. As flow:

Fig 2.28: sign language characters

So, after hand detection and tracking, the classifier take the output from hand detector module as input for classifier and classify the sign of the hand to certain character. In this project, we used deep learning classifier based on Keras to classify

hand sign characters. The model works on the bound boxes for hands detected in hand detector module, and for bound box it predicted the most match class for it. Our model had been trained to successfully recognize and classify 15 sign language characters, and that is very good in this case because Sign language is very difficult and its signs are somewhat similar.

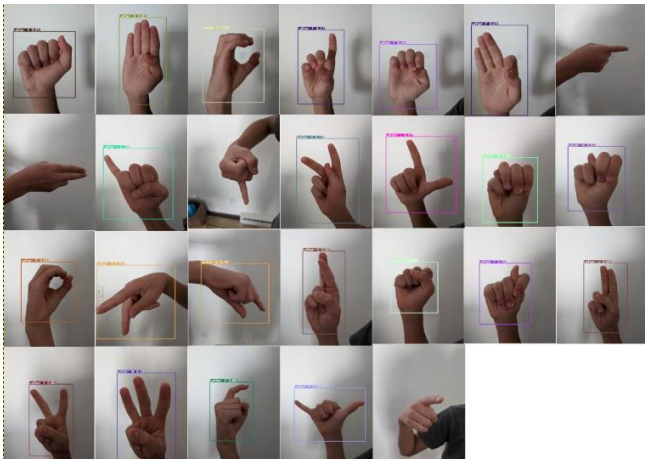


Fig 2.29: results of sign language recognition

Chapter 4

Used Software

4.1 Introduction

In this chapter we will discuss the software used in each part in the project and how they integrate with each other.

We can divide software used in the project to three main parts:

1. Maixduino Software
2. Jupiter notebook
3. Google colab

4.2 Maixduino Software

Here is software that used with MaixDuino board:

4.2.1 MaixPy IDE

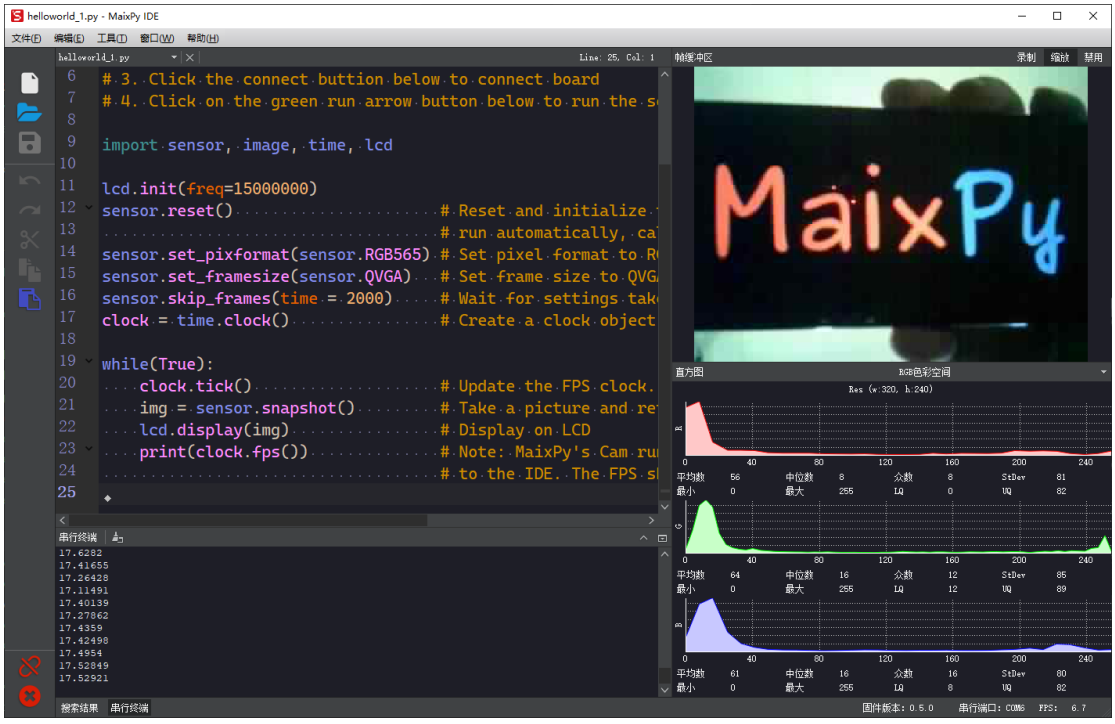


Fig 4.1: Maixpy IDE

First, you need to clarify: MaixPy uses Micropython script syntax, so it

does not need to be compiled like C language. In fact, it can be used happily without IDE: Use the serial terminal tool, which has been installed before

But using IDE will facilitate real-time editing of scripts on the computer and upload to the development board, execute scripts directly on the development board, and view camera images on the computer in real time, save files to the development board, etc.

Of course, the use of IDE will consume some resources for compression and transmission, so the performance will be reduced, and if MaixPy is down, there is no serial terminal to find the problem.

You can download maixpy IDE from this link:

https://cn.dl.sipeed.com/shareURL/MAIX/MaixPy/ide

4.2.2 MaixPy firmware

many firmware versions were introduced and compiled, so the smaller the firmware, the more libraries it takes. In STG project we will use ***maixpy_v0.6.2_84_minimum_speech_with_ide_support*** firmware.

Note that to use MaixPy IDE, the firmware must be v0.3.1 or higher, otherwise MaixPy IDE will not be connected.

4.2.3 Kflash_gui

With kflash_gui we can download the firmware to the development board.

Open the kflash_gui application, then select your firmware, setting options, and click to download. For more feature introduction and usage instructions, please refer to kflash GUI project homepage

When using, please note that the serial port cannot be occupied by other software, select the correct development board and serial port number, You can appropriately reduce the baud rate and use the low-speed mode to improve the download success rate.

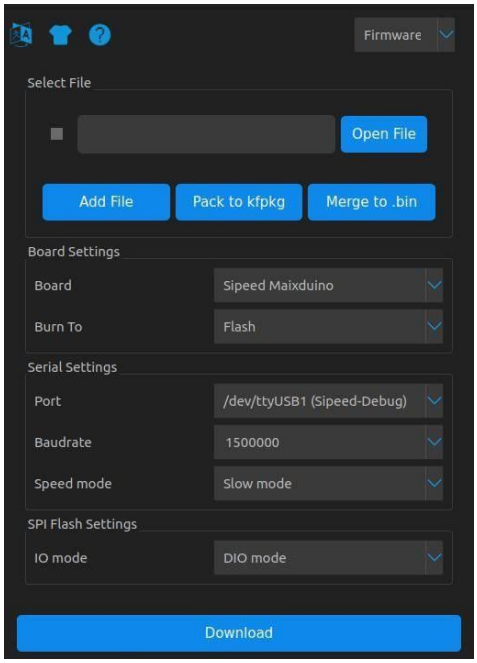


Fig 4.2: Kflash_gui

4.2.4 Connect Maixpy IDE with Maixduino

Open MaixPy IDE, select the model of the development board in the upper toolbar. Please select Maixduino to connect to amigo and cube development board.

Tool->Select Board

Choose Sipeed Maixduino board

Click 'connect' to connect to MaixPy IDE

After the connection is, the link button will change from green to red.

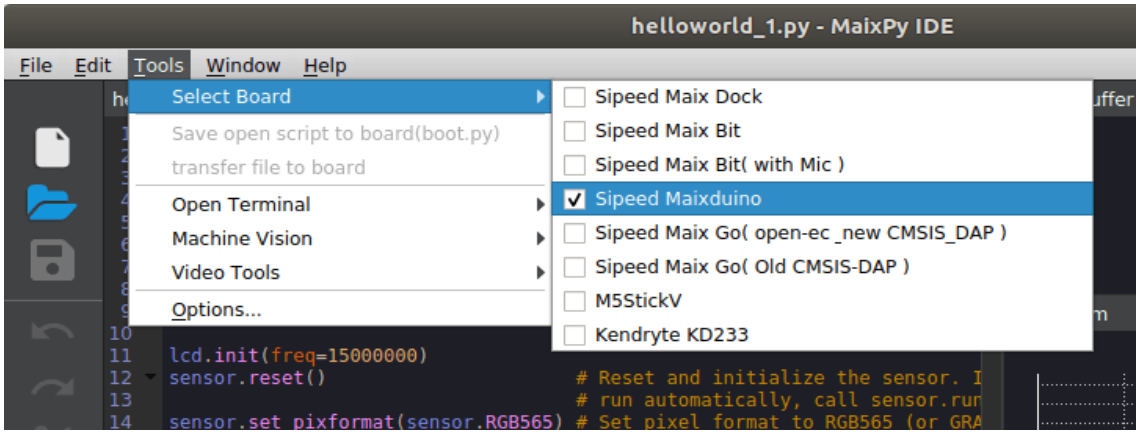


Fig 4.3: connect Maixpy IDE with Maixduino

4.2.5 Use MaixPy IDE

1. Open Maixpy IDE, connect the development board.

2. Edit the file, then in the top tools menu, use the function tool in the top (tools) menu to send the file, it will be saved to the development board and the file name is the same as the file name on the computer.

3. Of course, you can also click (boot.py) to save the code to (boot.py) the file of the development board, and the next time the development board is powered on, this file will be automatically executed.

Fig 4.4: Tools menu

4.2.6 Micropython: MaixPy project

Overview

MicroPython is a streamlined and efficient implementation of the programming language Python3. The syntax is consistent with Python3, but only a small part of the Python standard library is implemented, and it is optimized to be used in resource-constrained environments such as MCUs and WIFI SOCs.

MicroPython is packed full of advanced features such as an interactive prompt, arbitrary precision integers, closures, list comprehension, generators, exception handling and more. Yet it is compact enough to fit and run within just 256k of code space and 16k of RAM.

MicroPython has the same basic syntax of python3 like: Variables, Data types, Conditions, Loops, Functions and Modules.

Modules

As the code increases, the code in a file will become longer and harder to understand.

In order to write maintainable code, we group many functions into different files. In Python, a .py file is called a **module**.

The benefits of modules: Easy to reuse code! If I write a module and you write a module, we have two modules. We organize these modules, and everyone can write a lot less code.

How can we use modules: Modules can be imported through import the statement, ex: "import time "Just import time module.

The main modules used in the project:

- **Smart_tour_guide:** this module is not built-in module. It is made to call and control all function in the system. It is the main module in the system.
- **Time**: this module provides functions for getting the current time and date, measuring time intervals, and for delays.
- **lcd**: This module provides control of the MicroPython LCD160CR display.

- **gc**: This module implements a subset of the corresponding *CPython* module, as described below:
    - **gc.enable()** : Enable automatic garbage collection.
    - **gc.disable()** : Disable automatic garbage collection. Heap memory can still be allocated, and garbage collection can still be initiated manually using gc.collect().
    - **gc.collect()** : Run a garbage collection.
    - **gc.mem_alloc()** : Return the number of bytes of heap RAM that are allocated.

4.3 Jupyter Notebook

4.3.1 Overview

It is a web-based notebook environment for interactive computing in which you can combine code execution, rich text, mathematics, plots and rich media.

Jupyter Notebooks are primarily used by data professionals, particularly data analysts and data scientists. According to the Kaggle Survey 2022 results, Jupyter Notebooks are the most popular data science IDE, used by over 80% of respondents.

Fig 4.5: Jupiter logo

Fig 4.6: Jupiter-dashboard

the dashboard will give you access only to the files and sub-folders contained within Jupyter's start-up directory (i.e., where Jupyter or Anaconda is installed). However, the start-up directory can be changed.

4.3.2 Notebooks

A notebook integrates code and its output into a single document that combines visualizations, narrative text, mathematical equations, and other rich media. In other words: it's a single document where you can run code, display the output, and add explanations, formulas, charts, and make your work more transparent, understandable, repeatable, and shareable.

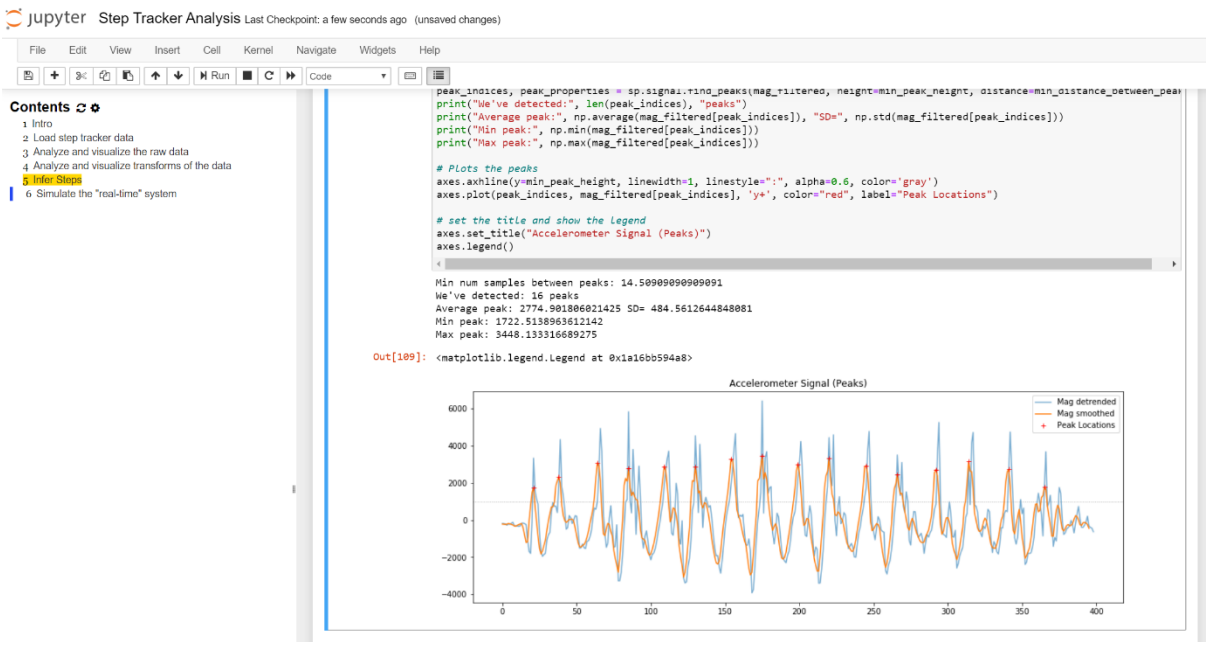


Fig 4.7: Jupiter notebook

Cells are the individual units of the notebook, and they can contain either text or code:

- Text cells are used to write narrative text and include images, links, and equations. Text cells are written in Markdown, a simple markup language.
- Code cells are used to write and execute code. The output from code cells will be displayed directly below the code cell.
- SQL cells (Workspace only) are used to execute SQL queries, which means you can easily retrieve data from a database.
- Chart cells (Workspace only) can be used to create visualizations and quickly visualize Panda's data frames.

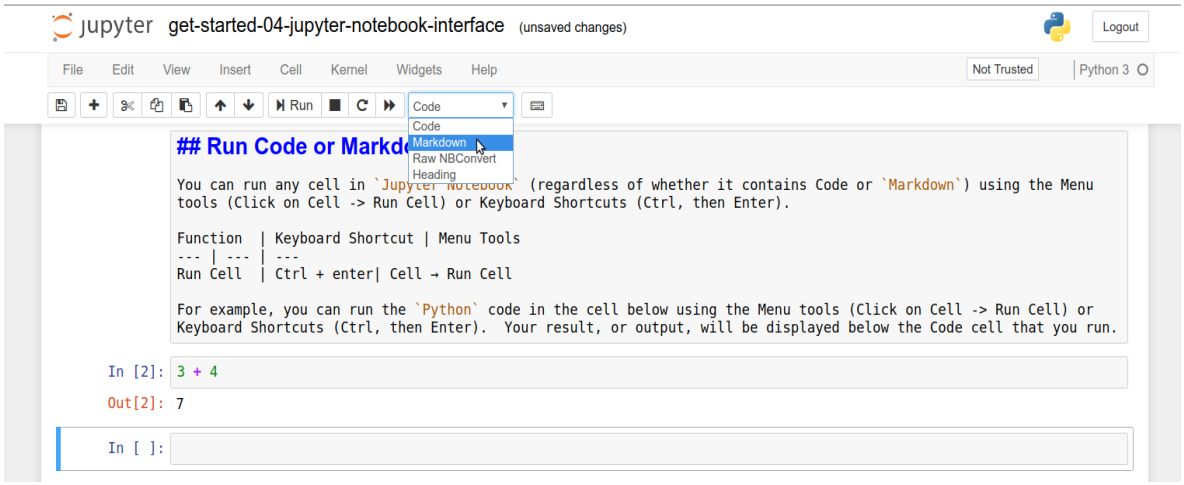


Fig 4.8: code and markdown cells

4.3.3 Main features of Jupiter Notebook

Originally developed for data science applications written in Python, R, and Julia, Jupyter Notebook is useful in all kinds of ways for all kinds of projects:

- **Data visualizations:** Most people have their first exposure to Jupyter Notebook by way of a data visualization, a shared notebook that includes a rendering of some data set as a graphic. Jupyter Notebook lets you author visualizations, but also share them and allow interactive changes to the shared code and data set.

- **Code sharing:** Cloud services like GitHub and Pastebin provide ways to share code, but they're largely non-interactive. With a Jupyter Notebook, you can view code, execute it, and display the results directly in your web browser.
- **Live interactions with code:** Jupyter Notebook code isn't static; it can be edited and re-run incrementally in real time, with feedback provided directly in the browser. Notebooks can also embed user controls (e.g., sliders or text input fields) that can be used as input sources for code.
- **Documenting code samples:** If you have a piece of code and you want to explain line-by-line how it works, with live feedback all along the way, you could embed it in a Jupyter Notebook. Best of all, the code will remain fully functional—you can add interactivity along with the explanation, showing and telling at the same time.

4.4 Programming

The main language used in this project is Python. Python's open-source libraries are not the only feature that make it favorable for machine learning and AI tasks. Python is also highly versatile and flexible, meaning it can also be used alongside other programming languages when needed. Even further, it can operate on nearly all OS and platforms on the market.

Implementing deep neural networks and machine learning algorithms can be extremely time consuming, but Python offers many packages that cut down on this. It is also an object-oriented programming (OOP) language, which makes it extremely useful for efficient data use and categorization.

4.5 Libraries

4.5.1 OpenCV

OpenCV is one of the image processing libraries in python. OpenCV provides a wide range of computer vision algorithms and functions, including image and video processing, feature detection, object recognition, machine learning, and more. It has a large community of developers and users, making it a popular choice for computer vision projects.

The image processing library provides access to over 2,500 state-of-the-art and classic algorithms. Users can use OpenCV to perform several specific tasks like removing red eyes and following eye movements.

Some of the features of OpenCV include:

- Image and video capture and processing
- Object detection and tracking
- Facial recognition and detection
- Optical character recognition (OCR)
- Machine learning algorithms for image and data analysis
- 3D reconstruction and augmented reality

OpenCV is easy to install and use. It comes with pre-built libraries and is compatible with multiple platforms such as Windows, Linux, macOS, iOS, and Android. It also has bindings for various programming languages like Python, Java, and MATLAB.

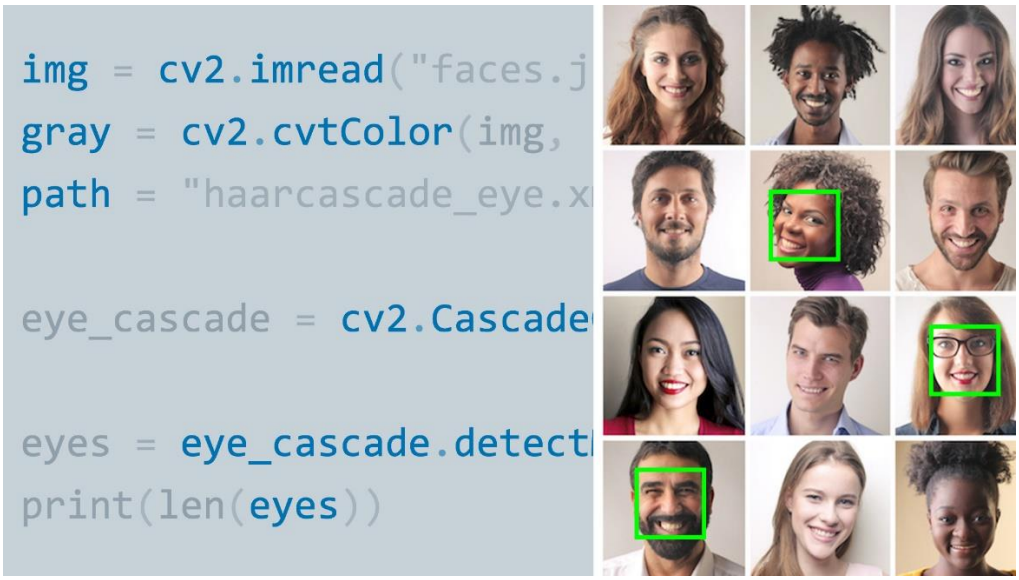


Fig 4.9: Face detection using OpenCV library.

4.5.2 Face recognition

Face recognition is one of the image processing libraries in python specified with face detection and recognition. Face recognition library is Built using dlib's state-of-the-art face recognition built with deep learning. The model has an accuracy of 99.38% on the Labeled Faces in the Wild benchmark.

Features:

- Find faces in pictures.

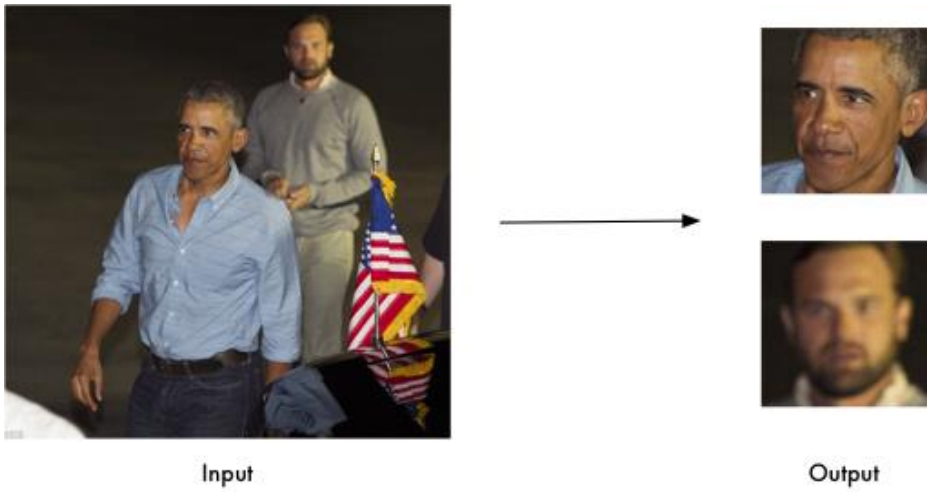


Fig 4.10: Face recognition

- Find and manipulate facial features in pictures.

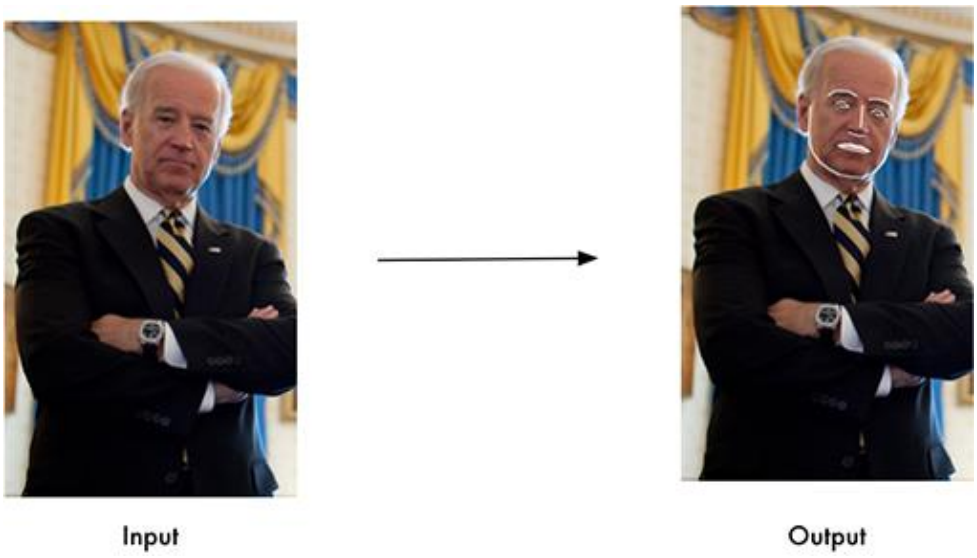


Fig 4.11: manipulate facial features in pictures.

4.5.3 Speech Recognition

Speech Recognition incorporates computer science and linguistics to identify spoken words and converts them into text. It allows computers to understand human language.

Speech recognition is a machine's ability to listen to spoken words and identify them. Speech recognition in Python can be used to convert the spoken words into text, make a query or give a reply. we can even program some devices to respond to these spoken words. we can do speech

recognition in python with the help of computer programs that take in input from the microphone, process it, and convert it into a suitable form.

Fig 4.12: Speech recognition

Speech recognition in Python works with algorithms that perform linguistic and acoustic modeling. Acoustic modeling is used to recognize phenomes/phonetics in our speech to get the more significant part of speech, as words and sentences.

Speech recognition starts by taking the sound energy produced by the person speaking and converting it into electrical energy with the help of a microphone. It then converts this electrical energy from analog to digital, and finally to text.

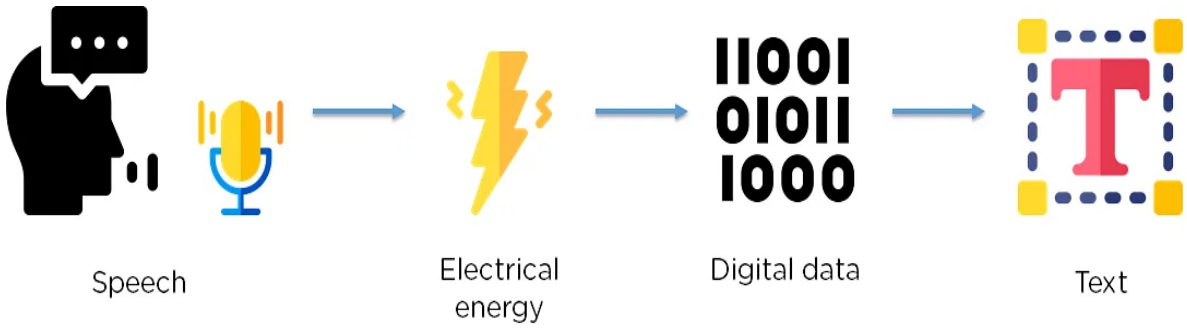


Fig 4.13: How speech recognition work

It breaks the audio data down into sounds, and it analyzes the sounds using algorithms to find the most probable word that fits that audio. All of this is done using Natural Language Processing and Neural Networks. Hidden Markov models can be used to find temporal patterns in speech and improve accuracy.

So, Speech Recognition package allows:

- Easy speech recognition from the microphone.
- Makes it easy to transcribe an audio file.
- It also lets us save audio data into an audio file.
- It also shows us recognition results in an easy-to-understand format.

4.3.5 pyttsx3

pyttsx3 is a text-to-speech conversion library in Python. Unlike alternative libraries, it works offline, and is compatible with both Python 2 and 3.

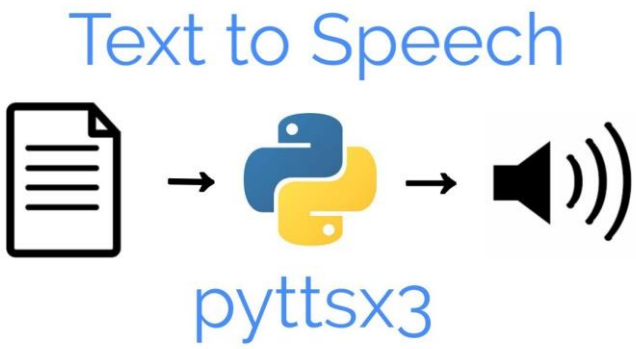


Fig 4.15: Text to Speech using pyttsx3 library.

4.3.5 TensorFlow and Keras

TensorFlow and Keras are two popular open-source libraries for building and training machine learning models in Python. TensorFlow was developed by Google Brain team, while Keras was developed by François Chollet, a Google engineer.

TensorFlow is a powerful and flexible library that allows you to create and train complex machine learning models. It provides a wide range of tools for building neural networks, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and more. TensorFlow also supports distributed computing, making it suitable for large-scale machine learning projects.

Keras, on the other hand, is a high-level API for building and training deep learning models. It provides a simple and intuitive interface for building neural networks and supports multiple backends, including TensorFlow, Microsoft Cognitive Toolkit (CNTK), and Theano. Keras is designed to be user-friendly and easy to learn, making it a popular choice for beginners.

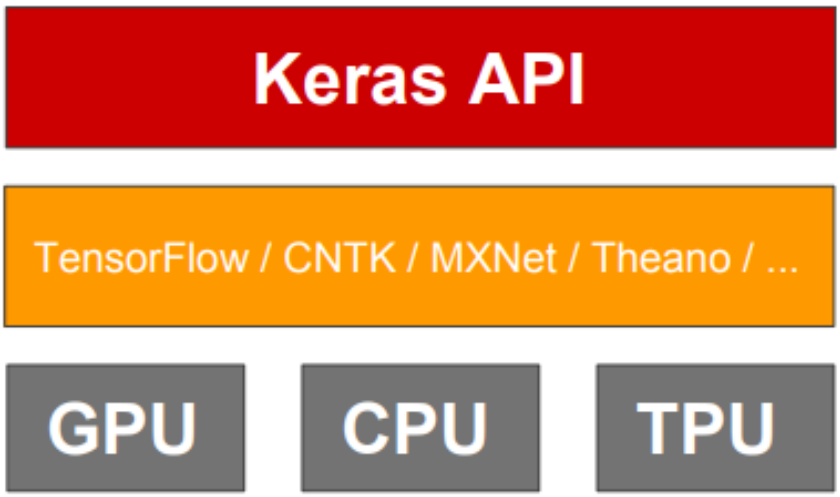


Fig 4.16: Keras and TensorFlow

One of the benefits of using Keras with TensorFlow is that it simplifies the process of building and training deep learning models. Keras provides a simple and intuitive interface for building and training neural networks, while TensorFlow provides the backend for executing the computations. This allows you to quickly prototype and test your models, without having to worry about the low-level details of the TensorFlow API.

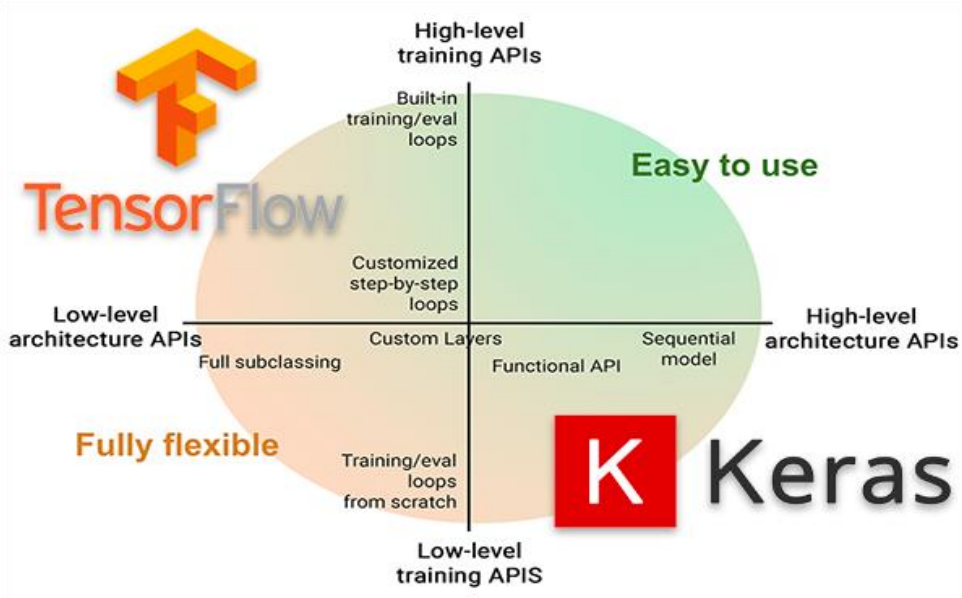


Fig 4.17: difference between Keras and TensorFlow

4.3.6 Cvzone

The `cvzone` library is a Python library built on top of OpenCV and provides a set of utility functions for computer vision tasks. It is developed by Indian software developer, CodeWithHarry.

Some of the features of the `cvzone` library include:

- Face detection and recognition tools
- Pose detection and tracking tools
- Image and video processing functions
- Object detection and tracking tools
- Hand tracking and gesture recognition tools
- Virtual mouse and keyboard tools

The `cvzone` library is easy to use and can be installed using the pip package manager in Python. It is compatible with both Windows and Linux operating systems.

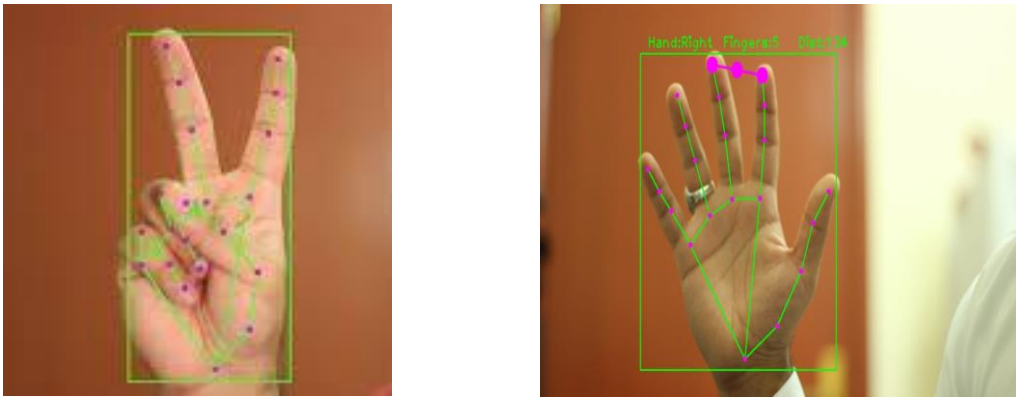


Fig 4.18: cvzone in hand detector

4.6 Yolo Algorithm

4.6.1 Overview

Object detection is a technique used in computer vision for the identification and localization of objects within an image or a video.

Image Localization is the process of identifying the correct location of one or multiple objects using bounding boxes, which correspond to rectangular shapes around the objects.

This process is sometimes confused with image classification or image recognition, which aims to predict the class of an image or an object within an image into one of the categories or classes.

You Only Look Once (YOLO) is a state-of-the-art, real-time object detection algorithm introduced in 2015 by Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi in their famous research paper "You Only Look Once: Unified, Real-Time Object Detection".

The authors frame the object detection problem as a regression problem instead of a classification task by spatially separating bounding boxes and associating probabilities to each of the detected images using a single convolutional neural network (CNN).

YOLO achieved state-of-the-art results, beating other real-time object detection algorithms by a large margin.

While algorithms like Faster RCNN work by detecting possible regions of interest using the Region Proposal Network and then performing recognition on those regions separately, YOLO performs all of its predictions with the help of a single fully connected layer.

Some of the reasons why YOLO is leading the competition include its:

5 Speed

6 Detection accuracy

7 Good generalization

8 Open source

4.6.2 YOLO Architecture

The YOLO algorithm takes an image as input and then uses a simple deep convolutional neural network to detect objects in the image. The architecture of the CNN model that forms the backbone of YOLO is shown below.

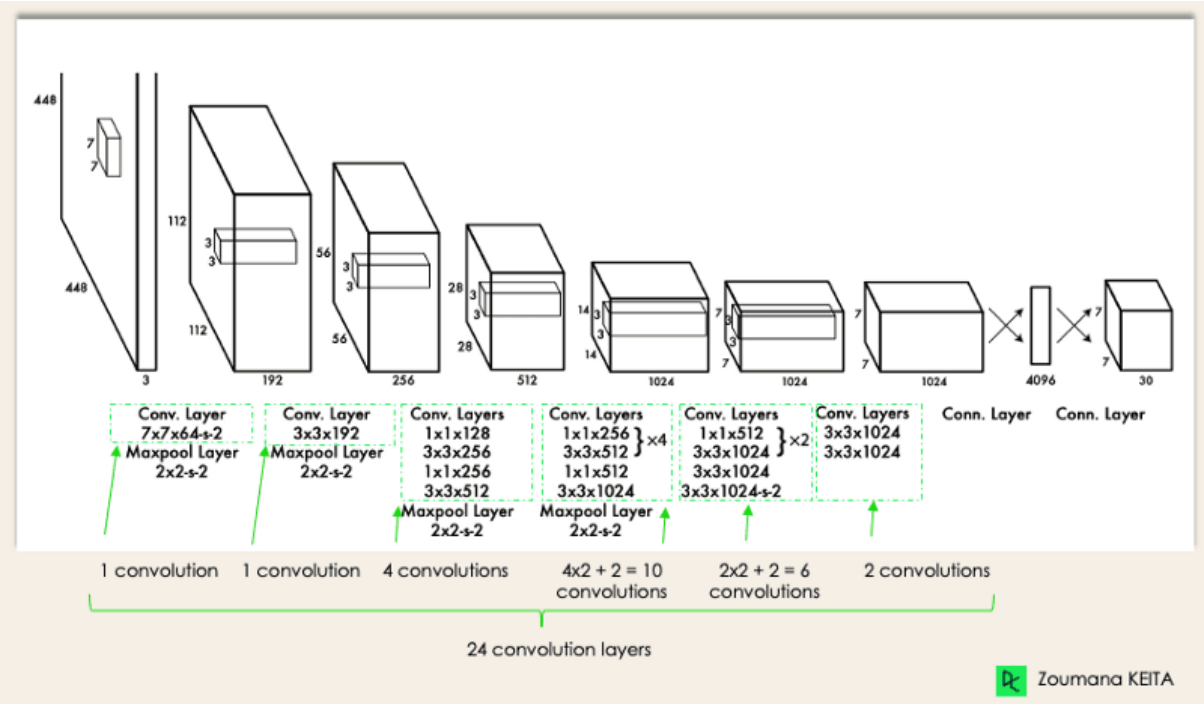


Fig 4.19: YOLO Architecture from the original paper

The first 20 convolution layers of the model are pre-trained using ImageNet by plugging in a temporary average pooling and fully connected layer. Then, this pre-trained model is converted to perform detection since previous research showcased that adding convolution and connected layers to a pre-trained network improves performance. YOLO's final fully connected layer predicts both class probabilities and bounding box coordinates.

4.6.3 How YOLO works?

The team that developed the algorithm apply a single neural network to the full image. This network divides the image into regions and predicates bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicated probabilities.

The image below shows that:

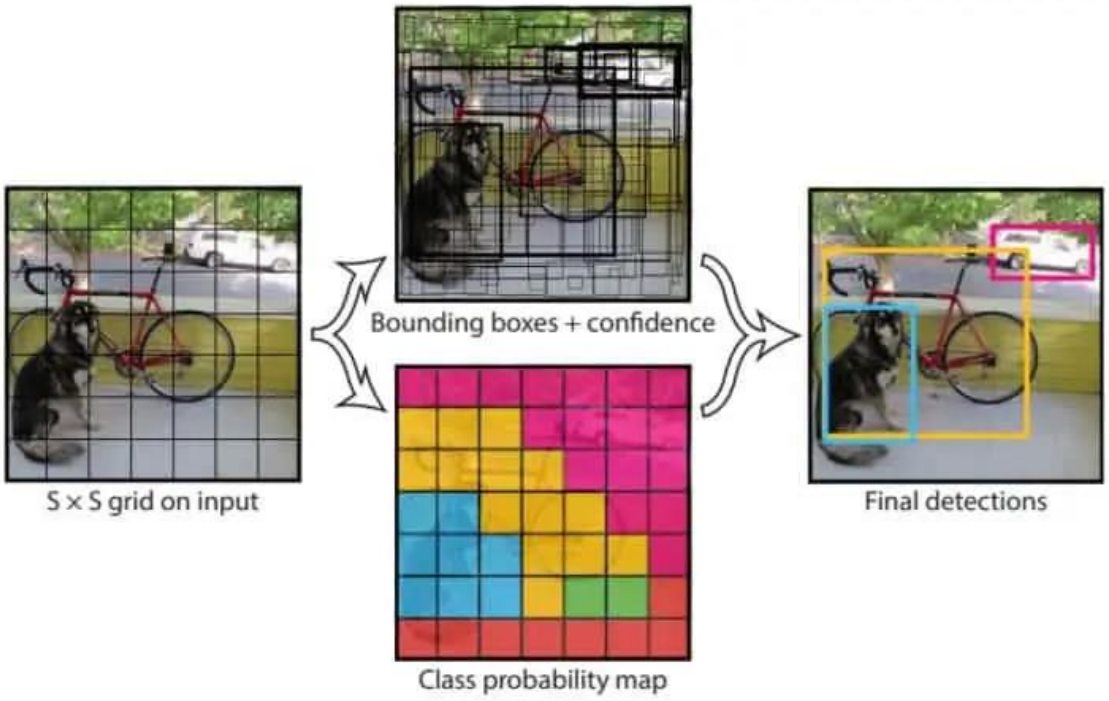


Fig 4.20: How YOLO works

4.6.4 YOLO v2.

YOLO v2, also known as YOLO9000, was introduced in 2016 as an improvement over the original YOLO algorithm. It was designed to be faster and more accurate than YOLO and to be able to detect a wider range of object classes. This updated version also uses a different CNN backbone called Darknet-19, a variant of the VGGNet architecture with simple progressive convolution and pooling layers.

One of the main improvements in YOLO v2 is the use of anchor boxes. Anchor boxes are a set of predefined bounding boxes of different aspect ratios and scales. When predicting bounding boxes, YOLO v2 uses a combination of the anchor boxes and the predicted offsets to determine the final bounding box. This allows the algorithm to handle a wider range of object sizes and aspect ratios.

Another improvement in YOLO v2 is the use of batch normalization, which helps to improve the accuracy and stability of the model. YOLO v2 also uses a multi-scale training strategy, which involves training the model on images at multiple scales and then averaging the predictions. This helps to improve the detection performance of small objects.

YOLO v2 also introduces a new loss function better suited to object detection tasks. The loss function is based on the sum of the squared errors between the predicted and ground truth bounding boxes and class probabilities.

The results obtained by YOLO v2 compared to the original version and other contemporary models are shown below.

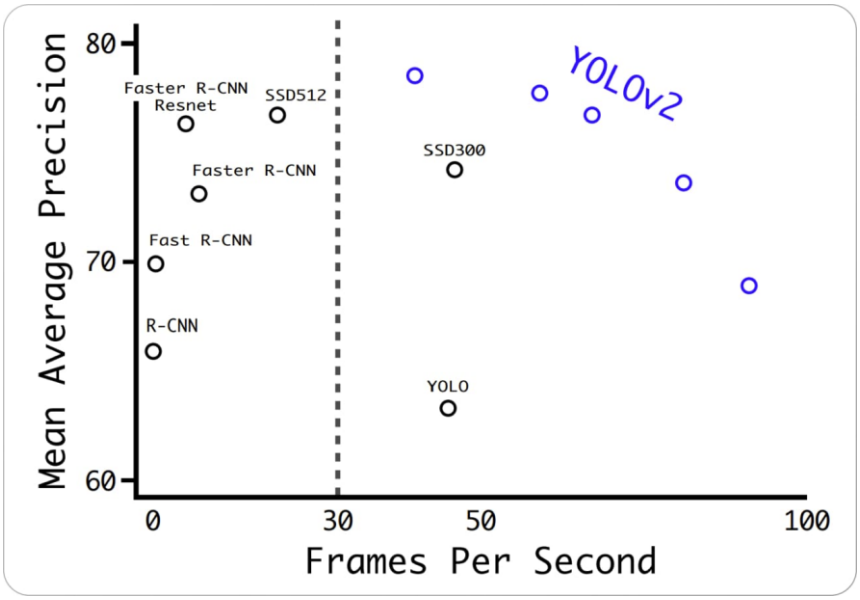


Fig 4.12: YOLO v2 compared to the original version and other contemporary models.

4.6.5 YOLO v5.

YOLO v5 was introduced in 2020 by the same team that developed the original YOLO algorithm as an open-source project and is maintained by Ultralytics. YOLO v5 builds upon the success of previous versions and adds several new features and improvements.

Unlike YOLO, YOLO v5 uses a more complex architecture called EfficientDet (architecture shown below), based on the EfficientNet network architecture. Using a more complex architecture in YOLO v5 allows it to achieve higher accuracy and better generalization to a wider range of object categories.

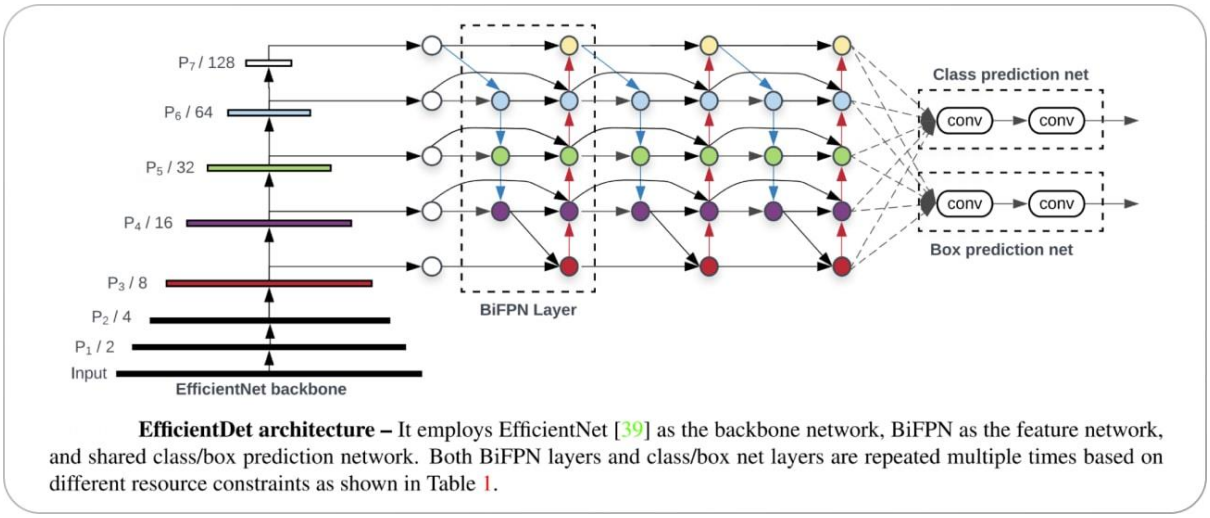


Fig 4.13: YOLOv5 architecture

Another difference between YOLO and YOLO v5 is the training data used to learn the object detection model. YOLO was trained on the PASCAL VOC dataset, which consists of 20 object categories. YOLO v5, on the other hand, was trained on a larger and more diverse dataset called D5, which includes a total of 600 object categories.

Chapter 5

System features

After connecting the hardware parts and testing the quality of all components, we will do implementation with our software part.

When the glasses are turned on, a welcome message will appear in front of you and a voice assistant informing you of the features of the glasses.

Fig 5.1: Welcome message in STG screen

In the following steps, we will tell you what some of the features offered by the glasses are and how to provide these features.

5.1 Face recognition

Here are five general steps that are typically involved in face recognition:

1. Data Collection: Gather a dataset of images containing faces. The dataset should include images of different individuals, captured from various angles, under different lighting conditions, and with different expressions. The more diverse and representative the dataset, the better the performance of the face recognition system.

2. Face Detection: Use a face detection algorithm or library to locate and extract faces from the images in your dataset. Face detection algorithms typically rely on techniques like Haar cascades, deep learning-based models (e.g., MTCNN, SSD), or OpenCV's built-in face detection functions.
3. Feature Extraction: Extract features from the detected faces to create a unique representation of each face. Traditional methods like Local Binary Patterns (LBP) or Histogram of Oriented Gradients (HOG) can be used, or you can leverage deep learning models like Convolutional Neural Networks (CNNs) to automatically learn and extract features.
4. Face Encoding: Convert the extracted features into a compact numerical representation called a face embedding or face encoding. This encoding captures the distinctive characteristics of each face. Popular techniques include using pre-trained CNN models (e.g., FaceNet, VGGFace) or dimensionality reduction methods like Principal Component Analysis (PCA) or Linear Discriminant Analysis (LDA).
5. Face Recognition: Compare the face encodings of new, unseen faces against the known face encodings in your dataset. Use similarity measures like cosine similarity or Euclidean distance to compute the similarity between face embeddings. If the similarity exceeds a predefined threshold, consider the face a match and recognize the individual. Otherwise, classify it as an unknown face.

We have done all these steps and below is the result:

If you use our glasses and talk to it and choose the face recognition function, it will open the camera for you and determine the image of the king in front of it in less than 5 seconds. If you press the device button, it will give you some information about this king.

Fig 5.1: Face recognition in Kings and Queens

In the following figure, I will show you how some information about the king recognized by the glasses is displayed.

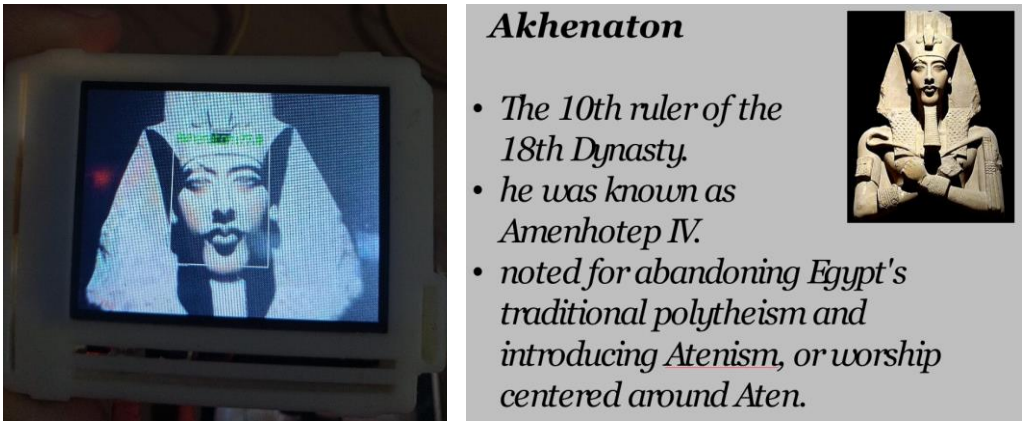


Fig 5.3: Face recognition in Tutankhamun and some information about him

5.2 QR Code

Here are five general steps that are typically involved in QR code:

1. Choose Data: Determine the type of data you want to encode in the QR code. It can be a URL, text, contact information, or other types of data.
2. Select a QR Code Generator: Use a QR code generator library or an online QR code generation tool. There are various libraries and websites available that offer easy-to-use interfaces for generating QR codes. You can choose one that suits your programming language or preferences.
3. Input Data: Provide the chosen QR code generator with the data you want to encode. Follow the instructions or API documentation of the generator to input the data correctly.
4. Customize Options: Optionally, customize the appearance and settings of the QR code. Many QR code generators allow you to modify the color, size, error correction level, and other parameters. Adjust these options according to your requirements.
5. Generate and Save: Generate the QR code using the generator. Once the QR code is generated, you can save it in a suitable format (such as PNG or JPEG) for use in your application or print materials.

If you choose the QR code function, the camera will open and read the qr code in front of you and display on the screen the contents of this code.

Fig 5.4: QR reader

5.3 Translation of ancient languages

Translating ancient languages can be a complex and specialized task that often requires expertise in linguistics, historical context, and the specific ancient language in question. While there is no one-size-fits-all approach, here are five general steps that are typically involved in the translation of ancient languages:

1. Language Proficiency: Develop proficiency in both the ancient language you are translating from and the modern language you are translating into. This involves studying the grammar, vocabulary, syntax, and writing systems of the ancient language. Additionally, understanding the linguistic and cultural context of the ancient civilization can be crucial for accurate translation.
2. Gather Resources: Collect relevant resources, such as dictionaries, grammars, linguistic analyses, and scholarly works on the ancient language. These resources serve as references to aid in understanding the vocabulary, grammar rules, and idiomatic expressions used in the ancient language.
3. Text Analysis: Analyze the ancient text that needs to be translated. Examine its structure, syntax, and any specific linguistic features. Identify key terms, idioms, or cultural references that may require special attention during translation.
4. Contextual Understanding: Develop a deep understanding of the historical, cultural, and social context in which the ancient text was written. This includes studying the time period, geographical location, and any relevant historical events, as well as the societal norms, customs, and religious or philosophical beliefs of the ancient civilization. This contextual knowledge is essential for capturing the intended meaning and cultural nuances in the translation.
5. Translation and Interpretation: Translate the ancient text into the modern language while considering the linguistic, cultural, and contextual factors. This step involves making informed choices regarding word choices, sentence structure, idiomatic expressions, and conveying the intended meaning of the original text accurately. It may also require interpreting ambiguous or uncertain passages based on the available information and expertise.

It's important to note that translating ancient languages can be challenging due to limited available resources, incomplete knowledge of the ancient language, and the potential for ambiguities and multiple interpretations.

Collaboration with experts in the field and continual research and learning are often necessary to refine and improve translations of ancient languages.

We have done all these steps and below is the result:

When user choosing the translate function, the camera opens and reads the visible symbols and determines what characters are in the image and translates them so that he can know the written name of the king and show it to the user.

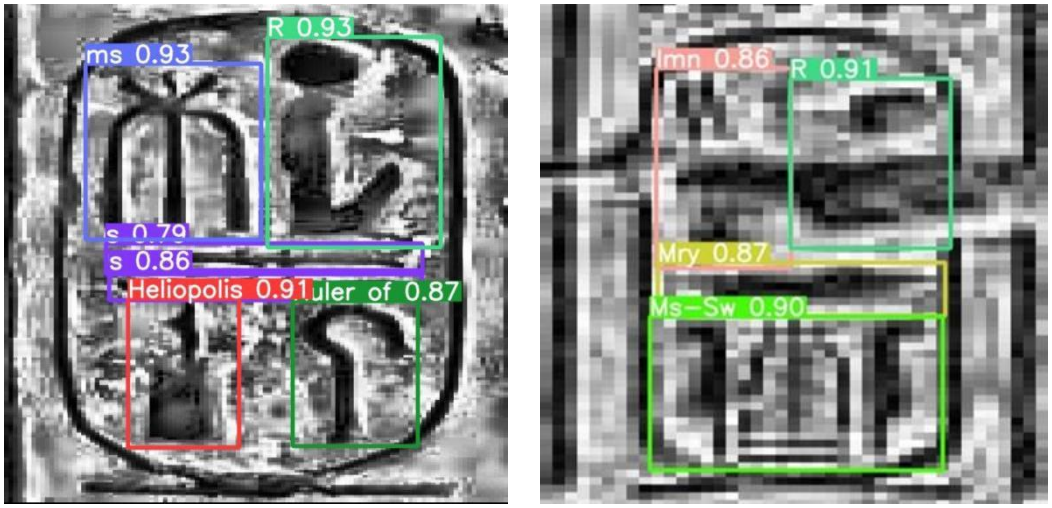
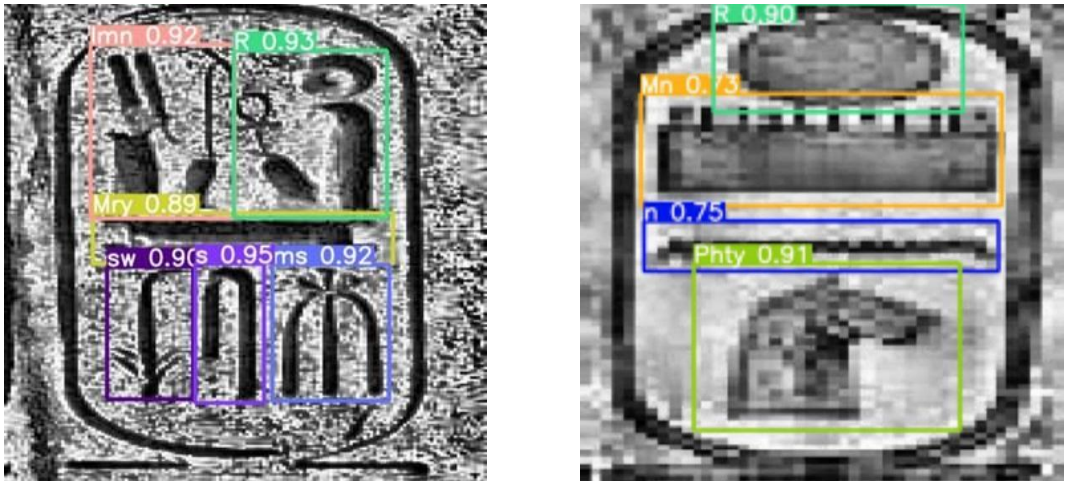


Fig 5.5: Detect the image to translate.



5.6: Accuracy for knowing words

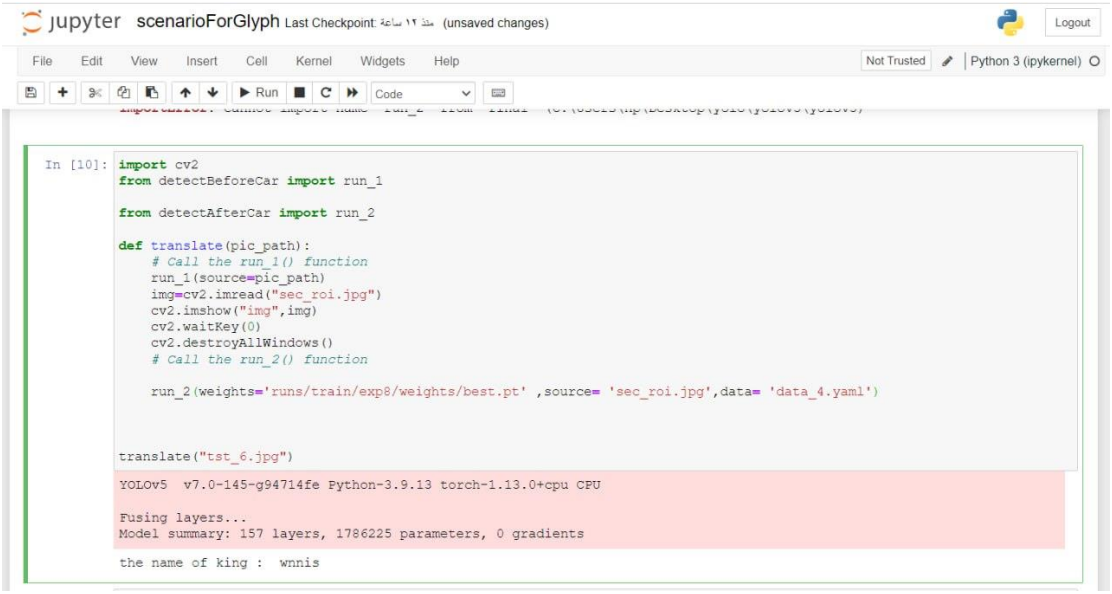


Fig 5.7: Code on Jupyter notebook about accuracy result

5.4 Assistant

Here are five general steps that are typically involved in voice assistant:

1. Speech Recognition: Implement a speech recognition system to convert spoken words into text. Use a speech recognition library or service that supports the programming language or platform you are working with. Popular options include libraries like Google's Speech-to-Text, Microsoft Azure Speech Service, or CMU Sphinx.
2. Natural Language Processing (NLP): Develop natural language processing capabilities to understand and interpret user commands. Apply techniques such as intent recognition, entity extraction, and sentiment analysis to extract meaning from the user's input. NLP frameworks like spaCy, NLTK, or the Natural Language Toolkit can assist in implementing these capabilities.
3. Task Execution: Implement the logic to perform tasks based on user commands. This can involve integrating with various APIs or services to accomplish specific actions, such as retrieving information from a database, making API requests, or executing predefined functions. Develop the necessary code and connectors to enable seamless task execution.
4. Text-to-Speech (TTS) Conversion: Implement a text-to-speech system to convert the assistant's responses into spoken words. Utilize a TTS library or service that can generate natural-sounding

speech from text. Popular options include libraries like Google's Text-to-Speech, Microsoft Azure Text-to-Speech, or open-source solutions like Festival or eSpeak.

5. User Interaction and Feedback: Design the user interaction and feedback loop for the voice assistant. Determine how the assistant will prompt the user for input, handle errors or misunderstandings, and provide relevant responses. Implement a dialogue management system to ensure a smooth conversation flow and handle different scenarios or user interactions.

We have done all these steps and below is the result:

When choosing the assistant or listening function, the microphone will open, and the user will enter any question he wants to know the answer to, whether about kings, museums, or any question related to archaeological monuments, etc. Then the program will take this question asked and search for it in Google, then it will show the answer to the question to the user.

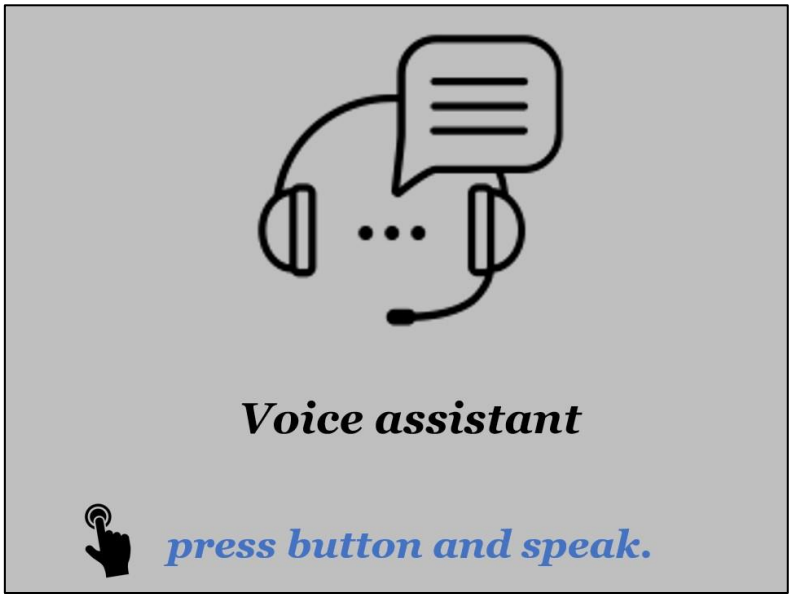


Fig 5.8: Voice Assistant message on screen

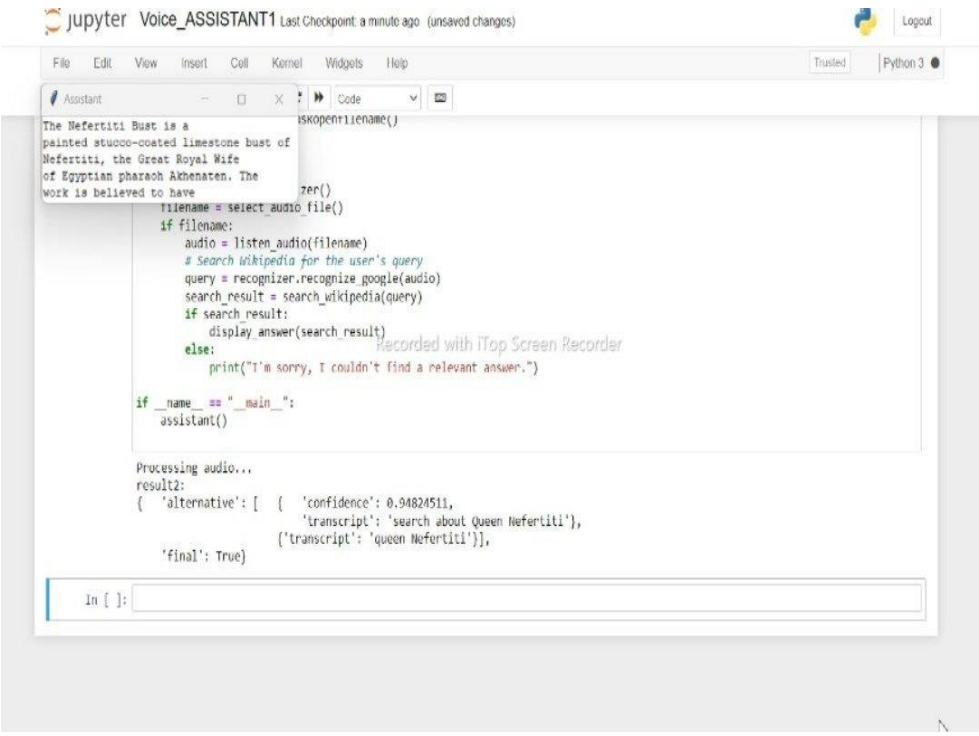


Fig 5.9: Code in Jupyter notebook showing information about Nefertiti.

It is also equipped with a language translation feature, so it can translate from one language to another, and it has a feature to know what the current weather is like. It can play some music, videos or audio clip.

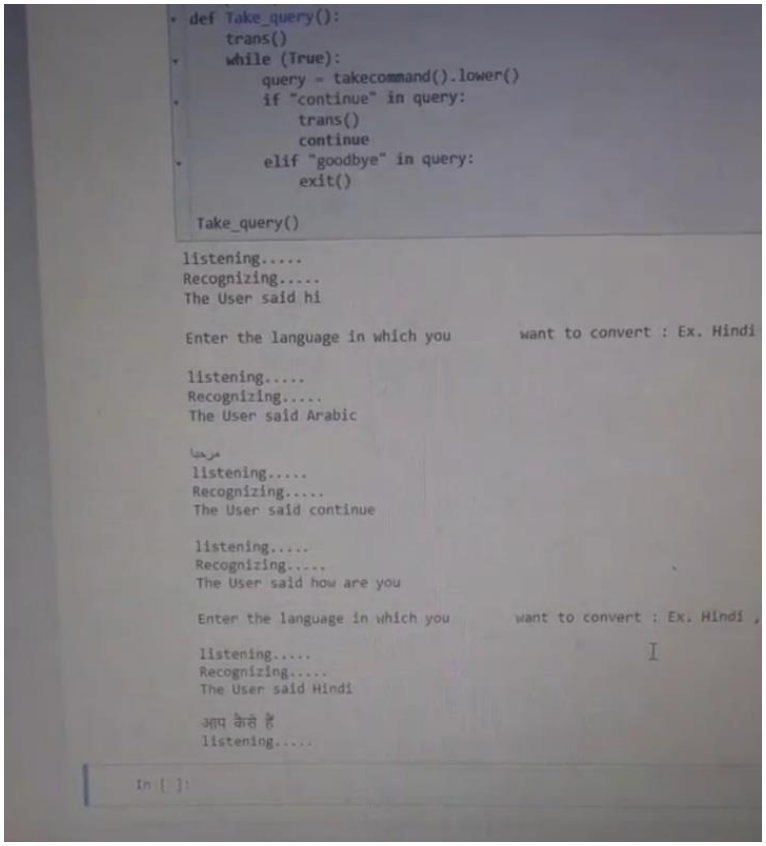


Fig 5.10: Translate Languages

5.5 Capture

When choosing the capture function, the camera will open, and user can take a memorial photo of himself and then print it.

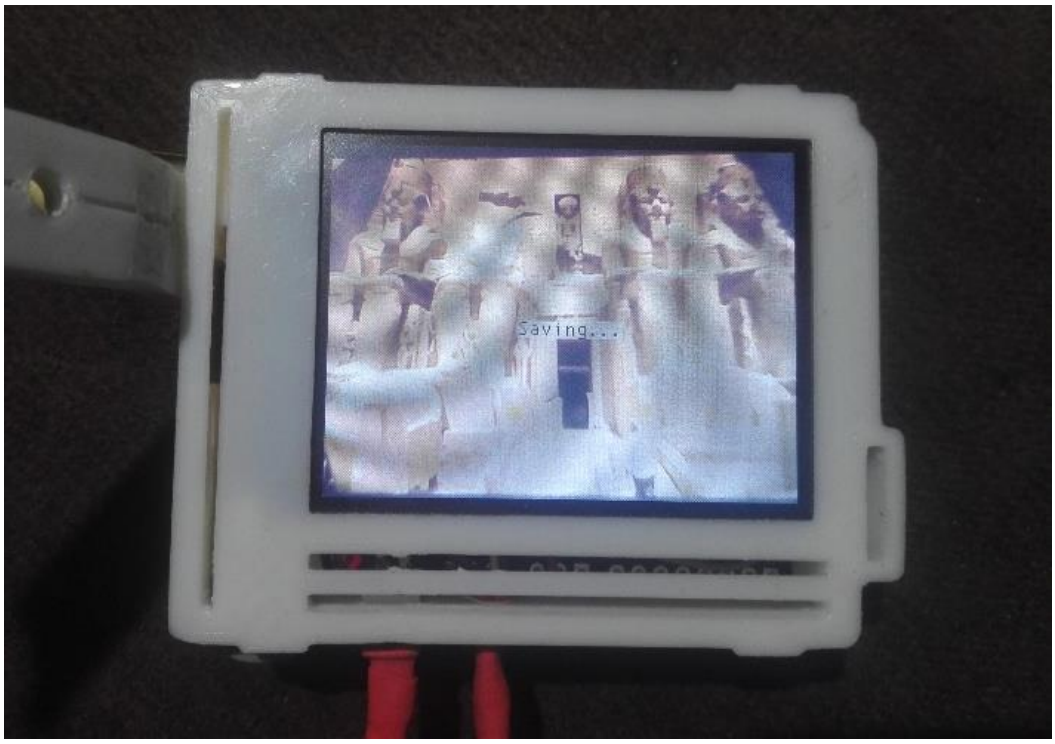


Fig 5.11: Capture photo

5.6 Sign language

In this project, we tried to cover some of the problems that People with hearing or audio special needs may face, so we trained her in sign language so that she could translate letters and words easily and she could deal with glasses without the help of another person.

In general, image recognition algorithms work by analyzing patterns in digital images and using those patterns to identify objects or features within the image. The process typically involves several steps:

1. Image acquisition: The first step is to obtain a digital image that will be analyzed by the algorithm. This can be done using a camera or by loading an existing image file.

2. Preprocessing: Before the image can be analyzed, it often needs to be preprocessed to reduce noise, adjust brightness and contrast, and perform other modifications to enhance the image quality.

3. Feature extraction: This step involves identifying specific patterns or features within the image that can be used to distinguish one object from another. This can involve using techniques such as edge detection, color analysis, or texture analysis.

4. Classification: Once the features have been extracted, the algorithm uses a machine learning model to classify the image based on those features. The model compares the extracted features to a set of known patterns or features that are associated with specific objects or categories.

5. Postprocessing: Finally, the algorithm may perform additional processing to refine the results, such as removing false positives or adjusting the confidence level of the classification.

Overall, image recognition algorithms rely on sophisticated machine learning techniques to analyze and classify digital images, and they have many practical applications, such as in object detection, facial recognition, and medical imaging.

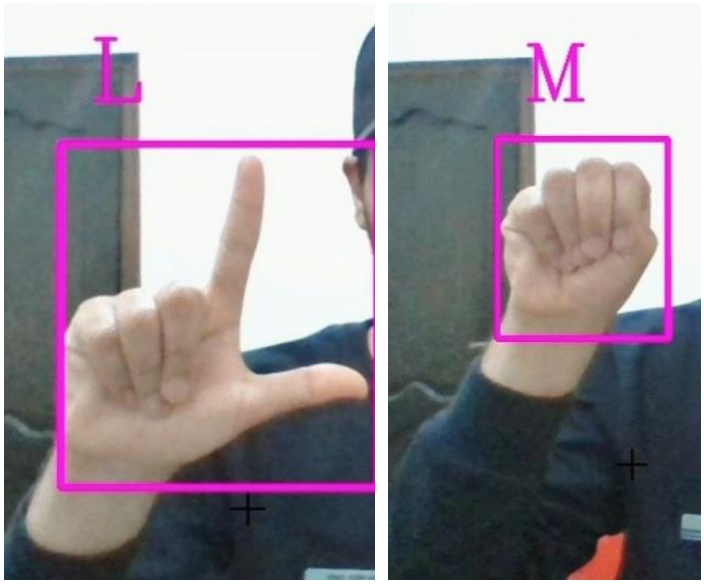


Fig 5.12: Sign language

5.7 Landmark detection

To perform landmark detection, which involves identifying and localizing specific landmarks or points of interest in an image, you can follow these five steps:

1. Data Collection: Gather a dataset of images that contain the landmarks you want to detect. This dataset should ideally cover different variations of the landmarks, including different viewpoints, lighting conditions, scales, and occlusions. The more diverse and representative the dataset, the better the performance of your landmark detection system.

2. Landmark Annotation: Annotate the images in your dataset by labeling the specific landmarks or points of interest you want to detect. This typically involves manually marking or annotating the landmark positions in each image. This step is crucial for training and evaluating your landmark detection model accurately.

3. Feature Extraction: Extract relevant features from the images to represent the landmarks. Commonly used techniques include using keypoint detectors like SIFT, ORB, or SURF to identify distinctive points, extracting local image descriptors, or leveraging deep learning-based feature extraction methods like convolutional neural networks (CNNs) to learn features directly from the images.

4. Training a Landmark Detection Model: Train a landmark detection model using the annotated dataset and the extracted features. This can involve using machine learning algorithms like support vector machines (SVM), random forests, or deep learning models like CNNs. The model should be trained to predict the landmark positions or keypoints based on the extracted features.

5. Landmark Detection and Localization: Apply the trained model to new images to detect and localize the landmarks. This involves

extracting features from the test images, feeding them into the trained model, and obtaining the predicted landmark positions. Post-processing techniques like non-maximum suppression or outlier rejection can be used to refine the predicted landmark positions if necessary.

We have done all these steps and below is the result:

We will consider this feature as sub project, and it is based on identifying the archaeological monuments in the museums and telling you their names, such as Abu Simbel Temple, the Sphinx,

Fig 5.13: Recognize Mohamed Ali Mosque with accuracy 0.84

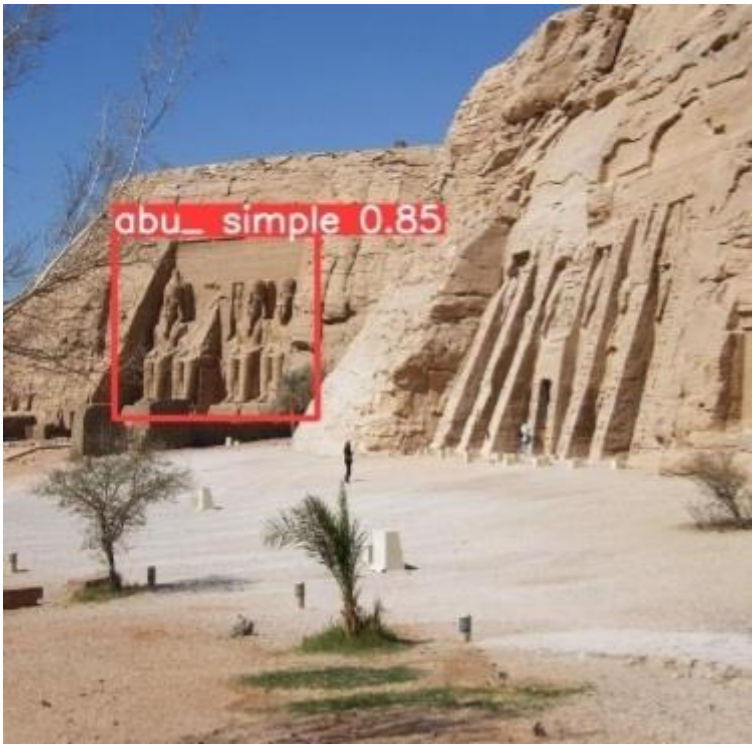


Fig 5.14: Recognize abu simple with accuracy 0.85

Fig 5.15: Recognize Hatshepsut with accuracy 0.68

Chapter 6

Conclusion & Future work

6.1 Conclusion

Smart tour guide has the possibility of enriching the tourism experience with the help of artificial intelligence.

Not only do smart glasses offer real-time navigation and translation capabilities, but they can also provide historical and cultural context to landmarks and attractions. This can greatly enhance the overall tourism experience by offering a deeper understanding of the places being visited.

- Our Smart glasses has some important features, including the personal assistant who takes the question from the user as a voice, converts the speech into text, performs a search on Google, and returns to the user the required answer as text.

- The second feature is the recognition of the face, as it identifies the faces of kings and pharaohs, and then gives them information about this king.

- The third feature is QR Code, as it has the ability to read the QR code or barcode in a simple way.

- The fourth feature is capturing photo, as it takes a souvenir photo of the tourist in high quality and sends this photo to the server so that you have the ability to print this photo.

- The fifth feature, which is the most important feature of our project, is that it can recognize texts written in hieroglyphs.

- And in anticipation of communicating with one of the disabled who is deaf and dumb, the sixth feature comes as the ability to recognize sign language.

- Last feature is Landmark detection is the process of finding significant landmarks in an image. We used it in our project in order to identify the archaeological monuments in the museum and talk about their names and some information about them.

Overall, while there are potential benefits and challenges associated with using smart glasses as a tour guide, they offer a unique and exciting way to enhance the tourism experience for both tourists.

6.2 Future work

- It will be possible to use it outside museums and tourist places so that you can move around and know everything through it.

- The possibility of having a conversation with the historical person using augmented reality technology.

- Using deep learning techniques, it will allow the tourist to see what this person looked like in reality.

- Mapping will be placed for each part of the museum so that the visitor can move around and know every corner of the museum or tourist antiquity without the help of anyone, by placing a map of the museum for the project.

- The project will be able to translate a complete text in hieroglyphs
- with ease so that we can read the ancient Egyptian texts.

Reference

[1.] Morris Franken and Jan C. van Gemert, "Automatic Egyptian Hieroglyph Recognition by Retrieving Images as Texts", in 2013, Intelligent Systems Lab Amsterdam (ISLA), University of Amsterdam Science Park 904 Amsterdam, The Netherlands.

[2.] Basem H. A. Ahmed, Ayman S. Ghabayen," Arabic Automatic Speech Recognition Enhancement", in 2017, Palestinian International Conference on Information and Communication Technology

[3.] Ankit Pandey, Vaibhav Vashist, Prateek Tiwari , Sunil Sikka, Priyanka Makkar, "Smart Voice Based Virtual Personal Assistants with Artificial Intelligence", in 2020 , Amity School of Engineering and Technology, Haryana, India.

[4.] Aman Preet Gulati, "Hand landmarks detection on an image using Mediapipe", in 2022, Analytics Vidhya

[5.] Ahmed K Elnagar , Abdelkader Mohammed Sghaier Derbali , "The Importance of tourism contributions in Egyptian economy", in September 2020, https://www.researchgate.net/publication/344013962_The_importance_of_tourism_contributions_in_Egyptian_economy

[6.] Simplilearn," Has Technology Improved Our Lives?", in 2023,https://www.simplilearn.com/how-has-technology-improved-our-lives-article .

[7.] What is Natural Language Processing? | IBM. (n.d.). https://www.ibm.com/topics/natural-language-processing.

[8.] IBM, "What is Computer vision?", What is Computer Vision? | IBM. (n.d.). https://www.ibm.com/topics/computer-vision.

[9.] Sipeed Wiki, "Introduction of MaixPy", in 2018, https://maixduino.sipeed.com/en/get_started/get_hardware.html.

[10.] Rohit Kundu,"YOLO: Algorithm for Object Detection Explained [+Examples]", in 2023, https://www.v7labs.com/blog/yolo-object-detection

[11.] Zoumana Keita,"YOLO Object Detection Explained", in 2022, https://www.datacamp.com/blog/yolo-object-detection-explained

[12.] kaspersky, "What is Facial Recognition – Definition and Explanation", https://www.kaspersky.com/resource-center/definitions/what-is-facial-recognition

[13.] Ahmed Hisham Misbah, "Can computer Vision recognize an ancient statue face?", in 2019, Research department at Artificial intelligence technology centerMisr university for Science and technology

[14.] Ali Tarhini, "Face Recognition: An Introduction", in 2010, https://alitarhini.wordpress.com/2010/12/05/face-recognition-an-introduction/

[15.] Firas Husham Al-Mukhtar,Mustafa Zuhaer Nayef Al-Dabagh, "Real-Time Face Recognition System Using KPCA, LBP and Support Vector Machine" in 2017 ,International Journal of Advanced Engineering Research and Science (IJAERS)

[16.] Benjamin Pryke,"How to Use Jupyter Notebook: A Beginner's Tutorial", in 2020, dataquest blog, https://www.dataquest.io/blog/jupyter-notebook-tutorial/

[17.] https://pypi.org/project/face-recognition/

[18.] MaixHub, “AI online training and sharing platform” https://maixhub.com/home

[19.] Maixduino example sketch "selfie" not working, in 2020, https://forum.arduino.cc/t/maixduino-example-sketch-selfie-not-working/667685

[20.] MaixPy_scripts on GitHub, in 2019, https://github.com/sipeed/MaixPy_scripts